Variational Models and Numerical Algorithms for Selective Image Segmentation

by

Lavdie RADA

under the supervision of:

Prof. Ke CHEN

UNIVERSITY OF LIVERPOOL

Thesis submitted in accordance with the requirements
of the University of Liverpool for the
degree of Doctor in Philosophy

May 2013

Contents

Acknowledgment

I would like to express my gratitude to all those people who gave me the possibility, helped and supported me to complete this thesis.

First, I would like to thank my supervisor Prof. Ke Chen for his great support, guidance, time and patience throughout my doctoral studies at the University of Liverpool.

I would like to thank other members of the mathematical sciences department for their advice and constructive criticism of my work: Prof. Alexander B. Movchan, Prof. Bakhti Vasiev, Dr. Yiqing Chen, Dr. Özgür Selsil and Dr. Kieran Sharkey.

I also would like to thank colleagues Dr. Carlos Brito, Dr. Behzad Ghanbari, Bryan Williams, Dr. Noppadol Chumchob, Dr. Jianping Zhang, Mazlinda Ibrahim, Jack Spencer and Dr. Yalin Zheng for all the interesting discussions we had during this time.

A very special thanks goes to Prof. Artan Borici, without whose motivation and encouragement I would not have considered a PHD career in mathematical research outside of my country.

My most sincere and deepest appreciation to my husband Şahan Ülgen, who patiently and lovingly supported me throughout my studies and is the reason for all my efforts. He made my life as graduate student much easier to handle and contributed to my enjoyment of this experience. Without his personal support I would not have been able to go through this long process.

I would also like to thank my family for the support they provided me through my entire life and in particular, I am very thankful to my parents, Elvete and Emin, who were my first teachers and allowed me to pursue my goals, and to my brother Kajo for his continuous support.

Last but by no means least, I recognize that this research would not have been possible without the academic and technical support of the Department of Mathematical Sciences, University of Liverpool and its staff, particularly in the award of a Postgraduate Research Studentship that provided the necessary financial support for this research.

Abstract

This thesis deals with the numerical solution of nonlinear partial differential equations and their application in image processing. The differential equations we deal with here arise from the minimization of variational models for image restoration techniques (such as denoising) and recognition of objects techniques (such as segmentation). Image denoising is a technique aimed at restoring a digital image that has been contaminated by noise while segmentation is a fundamental task in image analysis responsible for partitioning an image as sub-regions or representing the image into something that is more meaningful and easier to analyze such as extracting one or more specific objects of interest in images based on relevant information or a desired feature.

Although there has been a lot of research in the restoration of images, the performance of such methods is still poor, especially when the images have a high level of noise or when the algorithms are slow. Task of the segmentation is even more challenging problem due to the difficulty of delineating, even manually, the contours of the objects of interest. The problems are often due to low contrast, fuzzy contours, similar intensities with adjacent objects, or the objects to be extracted having no real contours.

The first objective of this work is to develop fast image restoration and segmentation methods which provide better denoising and fast and robust performance for image segmentation. The contribution presented here is the development of a restarted homotopy analysis method which has been designed to be easily adaptable to various types of image processing problems. As a second research objective we propose a framework for image selective segmentation which partitions an image based on the information known in advance of the object/objects to be extracted (for example the left kidney is the target to be extracted in a CT image and the prior knowledge is a few markers in this object of interest). This kind of segmentation appears especially in medical applications. Medical experts usually estimate and manually draw the boundaries of the organ/organs based on their experience. Our aim is to introduce automatic segmentation of the object of interest as a contribution not only to the way doctors and surgeons diagnose and operate but to other fields as well. The proposed methods showed success in segmenting different objects and perform well in different types of images not only in two-dimensional but in three-dimensional images as well.

Publications and Presentations

Publications

- L. Rada and K. Chen *A new variational model with dual level set functions for selective segmentation,* Communications in Computational Physics, Vol 12(1), pp.261-283 (2012).
- L. Rada and K. Chen and B. Ghanbari, *A restarted iterative homotopy analysis method for three-dimensional image segmentation*, International Journal of Signal and Imaging Systems Engineering, to appear.
- L. Rada and K. Chen, *On a variational model for selective image segmentation of features with infinite perimeter*, Journal of Mathematical Research with Applications, to appear.
- L. Rada and K. Chen, *A three-dimensional variational method for local segmentation,* Numerical Algorithms, to appear.
- B. Ghanbari, L. Rada and K. Chen, *A restarted iterative homotopy analysis method for two nonlinear models from image processing*, International Journal of Computer Mathematics, to appear.
- L. Rada and K. Chen, *Improved selective segmentation model using one level-set*, Journal of Algorithms and Computational Technology, submitted, 2012.
- L. Rada and K. Chen, *Improved selective segmentation model using one level-set and its viscosity solution*, SIAM Journal on Imaging Sciences, submitted, 2013.

Presentations

- **L. Rada** and K. Chen and B. Ghanbari *A restarted iterative homotopy analysis method for three-dimensional image segmentation*, IPTA'12, Istanbul (TR), October 15th -18th, 2012.
- **L. Rada** and K. Chen, *Improved selective segmentation model using one level-set*, EGH2012 Workshop on Applied and Numerical Mathematics, Greenwich (UK), June 7th -8th, 2012.

- **L. Rada** and K. Chen, *A variational infinite perimeter model for image selective segmentation,* SIAM Conference on Imaging Science (IS12), Philadelphia (USA), May 19th-23rd, 2012.
- **L. Rada** and K. Chen, *Selective segmentation and its application in tumour volume measurement,* Brain Tumour North West 5th Annual Retreat, Preston (UK), December 8th, 2011.
- **L. Rada** and K. Chen, *Selective segmentation in image processing,* MAGIC Postgraduate Conference, Reading (UK), January 12th-14th, 2011.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Image Processing and Variational Modeling

Nowadays computer vision, especially image processing, plays an increasingly important role in diverse subjects such as medical imaging, geophysics, geodesy, atmospheric science, medicine, biology, engineering, photography, film and video production, remote sensing, security monitoring etc. After modern photography was invented during the 18-th century by Louis-Jacques-Mandé Daguerre, which improved the process already established by Joseph-Nicéphore Niépce, the invention of charge-coupled device (CCD) opened the path for digital photography development and allowed not only storage but also computer processing for the images. In 1895, X-ray was invented,[1] and in the 1970s X-ray computed tomography (CT) becomes an important tool in medical imaging. A CT scanner uses ionizing radiation (X-ray) to obtain the image data. In 1977, another type of image, which was based on the emission and absorption of electromagnetic energy in the radio frequency (RF) range of the electrostatic spectrum was introduced by Paul Lauterbur and Peter Mansfield, is called the magnetic resonance imaging (MRI) scanner. These and other imaging tools are becoming more and more important in the modern world as a source of diagnosing illnesses, catching criminals, postmortal identification, etc.

Digital images are proper 2-dimensional (2-D) projections representations of the visual world surrounding us containing various objects. A 2-D digital image can be presented as a 2-dimensional array of data $z(x, y)$, where (x, y) represent the pixel location [2]. The pixel value corresponds to the brightness of the image at location (x, y). Some of the most frequently used image types are binary, gray-scale and color images. Binary images has only two possible values for each pixel, black and white, where black is represented with the value 0 while white with 1. They are also referred to as 1 bit/pixel images. Gray-scale images, also called monochromatic, represents the brightness of the image by carrying only intensity information. Gray-scale images contains 8 bits/pixel data, which means 256 different intensities (i.e., shades of gray) to

[1] http://inventors.about.com/library/inventors/blxray.htm

[2] *pixel* is a single point in a graphic image

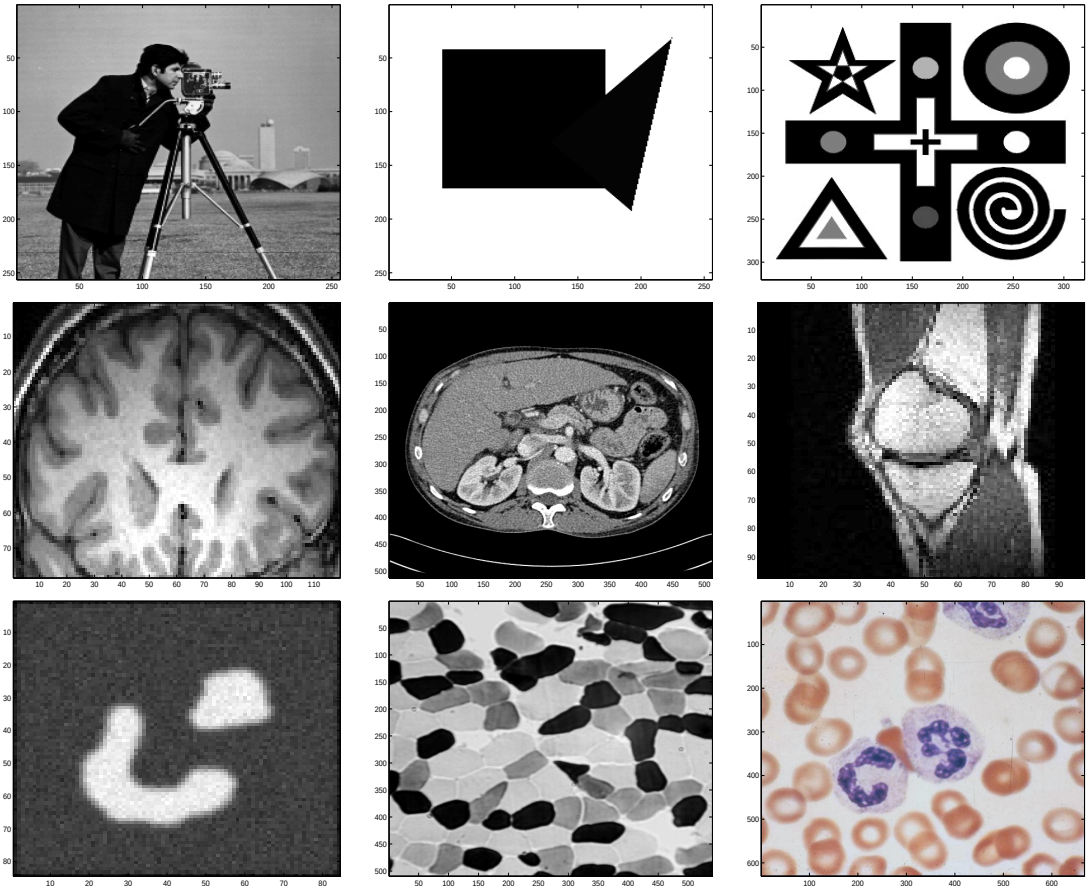


Figure 1.1: Sample images used in our experiments. Images in top row are real life and artificial images, the images in second row are MRI and CT images taken in hospital environments and images in third row are biological images of cells. The last image in this row is a color image.

be recorded between 0 (representing black) and 255 (representing white). Color images are considered as three-channel monochrome images, where one channel of information is dominated by red, another by green and the third by blue. The color image is produced by mixing together various proportions of red, green and blue lights. This is a 24 bits/pixel image and it is referred to as an RGB image. In discrete sense a grey image is a 2-D array of numbers (matrix) while a color image is a 2-D array of vectors (composed by three chancels). In continuous sense a grey image is a 2-D function $z(x, y)$ and color image is a 2-D vector function $(r(x, y), g(x, y), b(x, y))$. In this thesis we mainly work on grey value images. Fig. 1.1 shows some examples of images which will be used for our experimental work in coming chapters.

Technological advances in digital imaging, computer processors and mass storage devices fueled the growth of image processing. The huge volume of digital image data produced every day have become more than could ever be examined manually. Electronically perceiving and understanding the image and processing it according to specific human requirements has become a challenging task for computers and computer

programming. Therefore we have to develop mathematical models, algorithms and technologies with vision capabilities as advanced as human eyesight at least. Due to the wide specific application there is currently no single image processing method that yields acceptable results for every image and problem and for more some of the existing methods can at best deal with specific images only. Some of the techniques have been around already for some time and deeply investigated, though there is still room for improvement, others are relatively new and many challenges are still open for research.

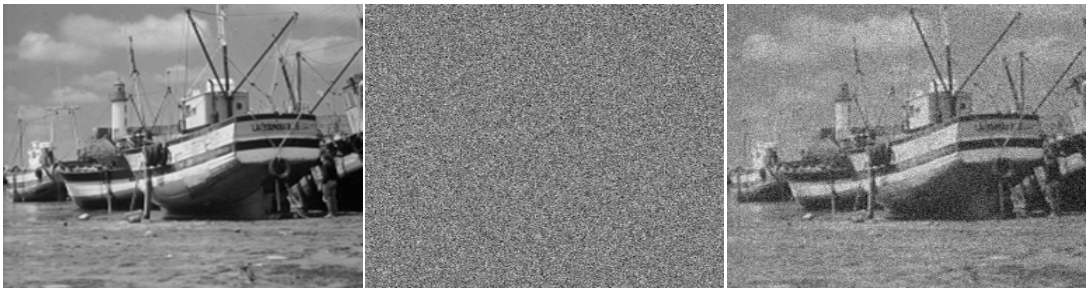
The field that includes methods for processing, analyzing, acquiring, and understanding images from the real world in order to produce numerical or symbolic information is called *Computer Vision*. Computer vision is divided into image processing, pattern recognition, motion analysis, statistical learning etc., which employs a range of more or less well-defined measurement problems or processing problems that can be solved using a variety of methods. The objectives are as varied as

- registration, mapping, comparing and combining two different views of the same object of a reference image into a target image,
- restoration and removal of noise (sensor noise, motion blur, etc.) from images,
- detection and recognition of objects in images,
- segmenting and picking out a feature of interest in an image from the rest of the image (the background),
- following the movements of a smaller set of interest points or objects in the image sequence (tracking of objects in videos),

and so on. In this thesis we look into a specific branch of image processing called *Image Denoising* and mainly into *Image Segmentation* which is a challenge and has various research open problems.

Due to the corruption of the real signals by noise (unwanted signals) in an image, during acquisition, transmission, and retrieval from storage media, images require preprocessing which removes the noise and restores the image as close as possible to the original image. Fig. 1.2 is an example of such an image which shows the photography of a boat taken with a digital camera.

The purpose of the denoising algorithm is to remove such noise. There are various methods to help restore an image from noisy distortions. Selecting the appropriate method plays a major role in obtaining the desired image. For example, a method that is used to denoise satellite images may not be suitable for denoising medical images. In this thesis we try to give a method which not only accurately restores degraded images but at the same time is an efficient solution in terms of speed performance. In order to quantify the performance of the various denoising algorithms, different images are taken and some known noise is added to it. Images are infected by different types of noise, as discussed in § 3.2. The work presented here focuses on a zero mean additive



(a) Clean Boat Image, z(x,y) (b) Noise, n(x,y) (c) Noisy Boat Image, u(x,y)

Figure 1.2: Illustration of noise in the boat image.

white Gaussian noise. The performance of each algorithm is compared by computing PSNR (the peak signal to noise ratio) or SNR (signal to noise ratio) besides the visual interpretation. Those measures are defined in Section §4.5.

The denoising problem can be mathematically presented as follows,

$$z(x, y) = u(x, y) + n(x, y) \tag{1.1}$$

where $z(x, y)$ is the observed noisy image, $u(x, y)$ the original image and $n(x, y)$ the noise with variance σ. The objective is to estimate $z(x, y)$ given $u(x, y)$.

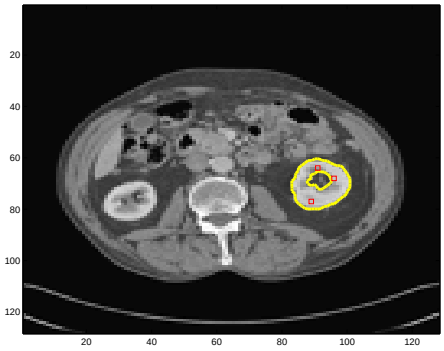
Image segmentation is about partitioning an image into multiple segments (sets of pixels) and disjoint sub-regions by modelling the similarity characteristic and common features of the desired object while dealing with variation in intensity, scale, pose, and shape. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images, to simplify and/or to change the representation of an image into something that is more meaningful. No single image segmentation technique performs well for all kinds of images, and problems; and in addition, the performance of various segmentation techniques is not the same and may vary from image to image. Segmentation should stop when the region of interest has been isolated. Due to this property the segmentation problem depends on the kind of the problem. In our work we have been dealing with the following problems for image segmentation:

- Develop a fast segmentation algorithm in order to extract the desired objects from the image background.
- Develop an automated selective/interactive segmentation method which has as an outcome a target object separated from other objects in the image.

Figure 1.3 shows two different segmentation problems where the first image is an example of image segmentation in which the boundaries of the objects are required to be segmented, shown as the contours of all letters in the the UoL image, and in the second image only the contour of the right kidney is the target object in a CT image.

A contour is introduced into the image to locate boundaries of features and is evolved

(a) UoL Image Segmentation



(b) Right Kidney Image Selective Segmentation

Figure 1.3: Illustration of segmentation of the UoL image and selective segmentation of a CT image where the right kidney is the object of interest.

until a steady state is reached, thereby we separate the objects from background or separate an object from other objects. The evaluation of the quality of segmentation outcomes is typically based on the visual inspection of the images and is critical to measure the efficiency of segmentation algorithms. The evaluation of the outcome of segmentation algorithms is made with human interaction.

For both the branches mentioned above as well as for image processing tasks in general, variational techniques are promising models to solve those problems. Finding the solution of variational models implies minimization of nonlinear functionals leading to numerical solution of Euler-Lagrange equations which are nonlinear partial differential equations (PDEs). Because of the discrete nature of the images, after using finite differences these PDEs are discretized and the models benefit from well-founded mathematical theories that allow us to analyze, understand and improve the existing methods. The Euler-Lagrange equation of these models is often described using parabolic PDEs, which are iterated in time until it reaches a steady state.

Numerical algorithms for image processing are known to be extremely slow, which is crucial nowadays since the size of the domain surpasses the million pixel mark and 3-dimensional (3-D) image processing is growing. Fast, stable and reliable numerical algorithms for this type of differential equations are required. This thesis deals with this problem by proposing fast, reliable and efficient numerical algorithms for the solution of nonlinear differential equations rising in image processing. The method we propose is a restarted homotopy analysis method, detailed in Chapter 4, which modifies the homotopy analysis method by making it easy and practical for hard nonlinear PDE cases. This method brings a better way to denoise and segment an image without introducing difficulties. Another advantage of the method is that it can easily be adapted to tackle all the nonlinear PDEs rising from image processing techniques in both 2-D and 3-D. The restarted homotopy analysis methods look attractive due to the low number of iterations used to obtain the steady state solution. For the image

denoising problem the restarted homotopy analysis method is applied to the Rudin-Osher-Fatemi denoising total variation equation [136, 135], which is a successful tool for image restoration and for Chan-Vese Active Contour Without Edges model [37] for image segmentation. The Chan-Vese model [37], which seeks the desired segmentation as the best piecewise constant approximation to a given image, is the most popular method in image segmentation. This method represents the contour with a zero level-set function originally developed by Osher and Sethian [122]. The basic idea is to start with initial boundary shapes, which in general are represented in the form of circular or rectangular closed curves, and iteratively modify them by applying shrink/expansion operations according to the constraints of the image. For both of these models, and similarly with other imaging processing models, the Euler-Lagrange equations associated are nonlinear PDEs. The restarted homotopy model will be compared with fast solvers such as the additive operator splitting method [174].

In the second part of the thesis we focus on variational image selective segmentation and introduce novel selective segmentation methods using the variational framework of active contours. More recently, Gout-Guyader [66, 69] and Badshah-Chen [12] proposed two different variational models for selective segmentation. The Badshah-Chen model improved the Gout-Guyader model, which is based on the edge information of the object, with a term which gives information to the minimization function of the region of segmentation. Both models can segment a range of images, but there are cases which appear too challenging for either model. For this reason we first develop a novel selective segmentation level-set method that uses a dual level set variational formulation. The model uses two level sets, from which one level set function (global) selects and finds all the objects, and with another level set (local) which is the closest to the geometric constraints (markers). It is a combination of edge detection, markers distance function and active contour without edges. Experimental results show that our model is more robust than previous work. To improve the efficiency of this model in the case of oscillatory boundaries a new regularization term has been proposed. The replacement of the regularizing term which measures the length of feature's boundaries with a $\mathcal{H}^1$ Hausdorff measure with $\mathcal{L}^2$ Lebesgue measure of the γ-neighborhood of the boundaries demonstrates the effectiveness of the proposed method for real life images with oscillatory boundaries. Even though the dual level set method is accurate the feedback of this method is a slow convergence in cases of large images or a 3-D process. For this reason we improve the one level set selective segmentation variational models by incorporating new terms of an adaptive parameter edge detection function and a new area-based fitting term which enhances the model's reliability.

1.2 Image Denoising and Image Segmentation

With the development of smart camera photo, high-resolution images, multi-slice medical imaging technology, telescopes and spectrographs, synthetic-aperture radar (SAR),

etc., methods of refining, acquiring, editing and visualizing image data are under a fascinating and ongoing improvement. Several applications of image processing such as in astronomy, astro-physics, biology, chemistry, art, genetics, physics, and other areas, bring a host of problems in imaging. The first and the most common one is the influence of noise which brings an inhomogeneous appearance of the surfaces or objects in an image. This is due to the limitation of the technology and low light illumination. Such problems can be easily faced in CT images, SAR images, smart camera images, etc. Although, the process of restoring the image has been widely investigated, there remain problems to be considered and better improvements are challenging.

A great challenge nowadays is to tackle automatically and identify intelligently objects in an image, in applications such as Closed-Circuit Television (CCTV) monitoring of a subject or medical geometry of a particular organ or tumors, where the selection of one feature/object among many ones is required. This is an important task which, particulary in the medical environment, will open the window to substantially improving patient diagnosis, treatment monitoring, and pre-operative planning. This will not only help us to view any organ on its own or as a collection within the anatomy, improving the diagnosis as well as surgical planning for patients, but could also allow the analysis of the shape and size of a tumor prior to surgery, or monitor the progress of a patient by comparing the segmented tumor through various stages of treatment. This will eliminate the manual segmentation which is a daunting task, since partial body scans range from tens to hundreds to even thousands of slices.

Due to the complexity of the images, especially the irregular shapes and sizes of objects/organs in medical images, the segregation process can be a difficult task. The edges can be difficult to delineate from the other organs due to noise during the image data acquisition process, or due to similar density with the other surrounding objects/organs.

There is a need for an automated process where some form of a priori information about the object such as markers are provided by the user. Chapters 5 and 6 provide a discussion on the novel 2-D and 3-D various models for image selective segmentation techniques currently available. These models are generally designed and are not complicated for any non-professional person to use and for more there is no need of manual manipulation control.

1.3 Chapters of this Thesis

The rest of this thesis is organized as follows:

Chapter 2

This chapter covers some basic mathematical tools which will be used throughout the rest of the chapters of the thesis and should prove helpful when reading and understanding them. Useful preliminary definitions, theorems and examples from normed

linear spaces, variations of a functional, bounded space of variations, ill-posed inverse problems, regularization for image processing and the level set methods will be shortly introduced. A discussion on the discretisation of partial differential equations (PDEs) on regular domains using finite difference methods and the iterative solutions of linear and nonlinear equations. An overview of implicit and explicit methods as well an introduction to fast solver algorithms, such as additive operator splitting and the homotopy analysis method, will be presented.

Chapter 3

This chapter is an introduction to variational models for image restoration and reconstruction techniques. The total variation (TV) regularization functional and some of its mathematical analysis and properties are introduced. Here we cover the TV denoising models based on the Mumford-Shah idea and in particular discuss the active contour without edges model of Chan-Vese. Some existing models used for solving the partial differential equation arising from Rudin-Osher-Fatemi (ROF) model and the minimization of the Chan-Vese model will be briefly discussed as well.

Chapter 4

A numerical discrete restarted homotopy analysis algorithm for solving the nonlinear partial differential equation of the TV model with applications in both denoising and segmentation is developed. This algorithm overcomes the nonlinearity of the TV model and at the same time gives a fast numerical scheme and a better method in terms of accuracy. Finally, numerical evidence will show the validity of the restarted discrete homotopy analysis method and that the method is efficient and robust even for images with large ratios of noise. The work has been generalised for 3-D as well and is shown by experiments to have great speed for 3-D segmentation. Moreover we will show that the method can be easily extended to other nonlinear TV models.

Chapter 5

This chapter presents a selective segmentation model using a dual level set variational formulation for a two dimensional selective model. This variational model aims to segment an object which is the closest to the geometric constraints (markers), with a local level set and a global level set which can serve as effective tools for identifying all features and their boundaries in an image. The model is a combination of edge detection, markers, a distance function and active contours without edges. An additive operator splitting method will be used to speed up the convergence of the algorithm. Experimental results show that our model is more robust than previous work.

Chapter 6

This chapter is a generalisation of the two dimensional dual level set selective model presented already in Chapter 5. The dual level set model developed in this chapter is capable of automatically capturing a local object of some target region in three dimensional domain. An additive operator splitting method is developed for accelerating the solution process. Numerical tests show that the proposed model is robust in locally segmenting complex image structures.

Chapter 7

This chapter improves the dual level set model, which is detailed in the previous two chapters, for irregular and oscillatory object boundaries in a selective model. The minimization energy replaces the Hausdorff measure of the length of a feature's boundaries (i.e. $\mathcal{H}^1$) with the Lebesgue measure of the γ-neighbourhood of the boundaries (i.e. $\mathcal{L}^2$). Experimental results show that in cases of real life images with oscillatory boundaries we get qualitative results demonstrating the effectiveness of the proposed method for these cases.

Chapter 8

In this chapter we propose a variational single level-set selective segmentation based model which is much faster to implement than the dual selective segmentation model in Chapter 5. The model combines several new ideas including a new area-based fitting term, a new region-based fitting term and an adaptive parameter edge detection function. The new model will be compared to the dual level set selective segmentation model and will be shown to have the same efficiency and reliability. We also provide an answer to the existence and uniqueness of the solution associated to our problem as well using viscosity theory. Test results show that the model finds the desired target object successfully in various challenging cases and that it is not heavily dependent on the prior information of markers or the distance function in contrast with existing selective models.

Chapter 9

In the last chapter we propose possible future research directions derived from the work presented in this thesis.

Chapter 2

Mathematical Preliminaries

This chapter covers some basic mathematical tools which will be used throughout the rest of the chapters of the thesis, helpful while reading and understanding them. Useful preliminary definitions, theorems and examples from normed linear spaces, variations of a functional, bounded space of variations, ill-posed inverse problems and regularization to image representation, the level set method and other commonly met methods in either linear algebra or advanced calculus literature will be introduced. A discussion of the discretization of partial differential equations (PDEs) on regular domains using finite difference methods and the iterative solutions of linear and nonlinear equations will be described.

We start by introducing the *Vector Space*, a basic mathematical structure formed by a collection of elements

$$\boldsymbol{u} = (u_1, \ldots, u_n), \quad \boldsymbol{v} = (v_1, \ldots, v_n). \tag{2.1}$$

called *vectors*, followed by the definitions of the *norm* and of *Normed Linear Spaces*. Literature can commonly be found in either linear algebra or advanced calculus literature such as [145, 88].

2.1 Normed Linear Spaces

Definition 2.1.1 (**Linear Vector Space**). *Let* $\mathbf{F}$ *be a scalar field (usually of real or complex numbers) and* $\mathbf{V}$ *a vector set on which two operations, addition and scalar multiplication, have been defined. For* $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{V}$*, the sum of* $\boldsymbol{u}$ *and* $\boldsymbol{v}$ *is denoted by* $\boldsymbol{u} + \boldsymbol{v}$*, and if* c *is a scalar, the scalar multiple of* $\boldsymbol{u}$ *by* c *is denoted by* $c\boldsymbol{u}$*. If the following axioms hold for all* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbf{V}$ *and for all scalars* $c, d \in \mathbf{F}$*, then* $\mathbf{V}$ *is called a vector space and its elements are called vectors.*
10 Axioms of a Vector Space:

1. *If* $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{V}$*, then* $\boldsymbol{u} + \boldsymbol{v} \in \mathbf{V}$ *(closure under addition)*
2. *If* $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{V}$*, then* $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u}$ *(commutativity under addition)*
3. *If* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbf{V}$*, then* $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ *(associativity of addition)*

4. *There exists an element* $\mathbf{0} \in V$ *, called a zero vector, such that* $\boldsymbol{u} + \mathbf{0} = \boldsymbol{u}$ *for all* $\boldsymbol{u} \in \mathbf{V}$ *(identity element of addition)*

5. *For each* $\boldsymbol{u} \in \mathbf{V}$*, there is an element* $-\boldsymbol{u} \in \mathbf{V}$ *such that* $\boldsymbol{u} + (-\boldsymbol{u}) = \mathbf{0}$ *(existence of additive inverse)*

6. *If* $c \in \mathbf{F}$*, and* $\boldsymbol{u} \in \mathbf{V}$*, then* $c\boldsymbol{u} \in \mathbf{V}$ *(closure under scalar multiplication)*

7. *If* $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{V}$*, and* $c \in \mathbf{F}$ *then* $c(\boldsymbol{u} + \boldsymbol{v}) = c\boldsymbol{u} + c\boldsymbol{v}$ *(distributivity)*

8. *If* $\boldsymbol{u} \in \mathbf{V}$*, and* $c, d \in \mathbf{F}$ *then* $(c + d)\boldsymbol{u} = c\boldsymbol{u} + d\boldsymbol{u}$ *(distributivity)*

9. *If* $\boldsymbol{u} \in \mathbf{V}$ *and* $c, d \in \mathbf{F}$ *then* $c(d\boldsymbol{u}) = (cd)\boldsymbol{u}$ *(associativity of scalar multiplication)*

10. *There exists an element* $1 \in \mathbf{V}$*, called the multiplicative identity, such that* $1\boldsymbol{u} = \boldsymbol{u}$ *for all* $\boldsymbol{v} \in \mathbf{V}$ *(identity of scalar multiplication)*

Definition 2.1.2 (**Norm**). *For a given a vector space* $\mathbf{V}$ *over a subfield* $\mathbf{F} \subseteq \mathbb{C}$*, a real valued function* $N : \mathbf{V} \longrightarrow \mathbb{R}$ *is called a* ***norm*** *on* $\mathbf{V}$ *if for all* $a \in \mathbf{F}$ *and all* $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{V}$*, it satisfies*

1. $N(\mathbf{v}) > 0$ *for all* $\mathbf{v} \neq 0 \in \mathbf{V}$ *and* $N(\mathbf{v}) = 0$ *for* $\mathbf{v} = 0$*, (separates points).*

2. $N(\alpha\mathbf{v}) = |\alpha| N(\mathbf{v})$ *for all* $\alpha \in \mathbb{R}$ *and* $\mathbf{v} \in \mathbf{V}$*, (triangle inequality).*

3. $N(\mathbf{v} + \mathbf{u}) \leq N(\mathbf{v}) + N(\mathbf{u})$ *for all* $\mathbf{v}, \mathbf{u} \in \mathbf{V}$*, (positive homogeneity).*

A **norm** is a **seminorm** if the 1-st property (separating points) is removed.
A norm on a vector space $\mathbf{V}$ induces a metric on $\mathbf{V}$ by

$$d(\mathbf{v}, \mathbf{u}) := N(\mathbf{v} - \mathbf{u}).$$

This metric is invariant under translations and homogeneity, i.e.

$$d(\mathbf{v} + \mathbf{w}, \mathbf{u} + \mathbf{w}) = d(\mathbf{v}, \mathbf{u}), d(\lambda\mathbf{v}, \lambda\mathbf{u}) = |\lambda| d(\mathbf{v}, \mathbf{u}).$$

The norm of a vector $\mathbf{x}$ on the set of real numbers $\mathbb{R}$ is usually represented by $\|\mathbf{x}\|$ (or for simplicity in some cases $|x|$), and the function is denoted $\|\cdot\|$ (or $|\cdot|$).

Example 2.1.3 *Some examples of norms*

- *The absolute value is a norm on the set of real numbers* $\mathbb{R}$*.*

- ***Euclidean norm of a vector:***
  Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ *then*

  $$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}.$$

  This gives the ordinary distance from the origin to the point $\mathbf{x}$*. Note that this norm can sometimes be written as* $\|\mathbf{x}\|_2$ *or* $|\mathbf{x}|$*.*

- ***Infinity norm:***
  For $\mathbf{x} \in \mathbb{R}^n$*,* $\|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \ldots, |x_n|)$*.*

- ***p-norm of a vector:***
  Consider $\mathbf{x} \in \mathbb{R}^n$*, then for any real number* $p \geq 1$ *the p-norm of* $\mathbf{x}$ *is defined as*

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \tag{2.2}$$

  For $p = 1$ *this norm is called the 1-norm and clearly, for* $p = 2$ *this is the Euclidean norm, and as p approaches* ∞ *the p-norm approaches the infinity norm.*

- ***L^p-norm of a function:***
  Consider a continuous function f *defined on a domain* Ω *such that* $\int_\Omega |f(x)|^p \, dx < \infty$ *with* $1 \leq p \leq \infty$*. Then*

$$\|f(x)\|_{L^p} = \left(\int_\Omega |f(x)|^p \, dx \right)^{1/p} \tag{2.3}$$

  defines the L^p*-norm of* f *on* Ω*. This is a generalization of the previous example since now* f *is allowed to have arbitrarily many components.*
  The special case when $p = \infty$ *is defined as*

$$\|f(x)\|_\infty = \sup_x |f(x)|. \tag{2.4}$$

- ***Total Variation (TV) norm*** *of* $u : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}$ *is defined as*

$$TV(u) = \int_\Omega |\nabla u| \, dxdy.$$

  and will be discussed in more detail later in this chapter.

Definition 2.1.4 (**Normed Linear Space**). *A vector space equipped with a norm (seminorm)* $\|.\|$ *defined on it is called a normed linear space (seminormed linear space).*

This also means that a linear vector space together with an inner product defined on it, is a special type of normed space.

Theorem 2.1.5 *For* $p \in [1, \infty]$*, the space equipped with the* L^p *norm is a normed vector space.*

Proof The proof of this theorem uses the content of Minkowski's inequality, Theorem 2.1.19, shown below. □

Definition 2.1.6 *The space of all n-tuples of real numbers,* $(x_1, x_2, ..., x_n)$ *commonly denoted* $\boldsymbol{x} \in R^n$*, equipped with the Euclidean metric*

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{n} (x_i - y_i) \right)^{1/2}$$

is called Euclidean n-space or Cartesian space.

Definition 2.1.7 (**Cauchy Sequence**). *A Cauchy sequence in a normed vector space* $\mathbf{V}$ *is a sequence* $\{\mathbf{x}_i\}_{i=1}^{\infty}$ *having the property that for any* $\varepsilon > 0$*, there exists an* $N \in \mathbb{N}$ *such that*

$$\|x_i - x_j\| < \varepsilon, \qquad \forall i, j \geq N.$$

Note that Cauchy sequence is a sequence where all the terms, except a counted number, become arbitrarily close to one another

Definition 2.1.8 (**Banach Space**). *A normed space* $\mathbf{V}$ *is said to be a Banach space if every Cauchy sequence* $\{x_i\}_{i=1}^{\infty} \subset \mathbf{V}$ *converges to an element* $x \in \mathbf{V}$ *(which means that* $\lim_{i\to\infty} x_i = x$*).*

Definition 2.1.9 (**Hibert Space**). *A Hibert space is a space* $\mathbf{V}$ *with an inner product* $\langle u, v\rangle$ *such that every Cauchy sequence converges to an element of the space* $\mathbf{V}$*.*

A Hilbert space is always a Banach space, but the converse need not hold.

Definition 2.1.10 (**Convex Set**). *A set* $\mathbf{S}$ *in a vector space* $\mathbf{V}$ *is said to be convex if, for all* $u, v \in \mathbf{S}$ *and all* $\theta \in [0, 1]$*, the point*

$$(1 - \theta)u + \theta v$$

is in $\mathbf{S}$*. In other words, every point on the line segment connecting* u *and* v *is in* $\mathbf{S}$ *.*

Definition 2.1.11 (**Convex Functions**). *A function* $f : \mathbf{S} \to R$ *defined on an convex set* $\mathbf{S}$ *of some vector space is called convex if*

$$f(\theta u + (1 - \theta)v) \leq \theta f(x) + (1 - \theta)f(y) \tag{2.5}$$

for all $x, y \in S$ *and* $\theta \in (0, 1)$*. If the inequality is always strict for* $u \neq v$*,* f *is called strictly convex.*

Example 2.1.12 *Examples on* $\mathbb{R}$ *and* $\mathbb{R}^n$

- *exponential:* e^{ax}*, for any* $a \in \mathbb{R}$ *on domain* $\mathbb{R}$ *is convex.*
- *powers:* x^{α} *on* $\mathbb{R}^+$*, for* $\alpha \geq 1$ *or* $\alpha \leq 0$ *is convex.*
- *powers of absolute value:* $|x|^p$ *on* $\mathbb{R}$*, for* $p \geq 1$ *are convex.*
- *affine function:* $f(x) = a^T x + b$ *where* $a \in \mathbb{R}^n, x, b \in \mathbb{R}^{n\times 1}$ *is convex.*
- *The norms:* $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$*, for* $p \geq 1$*;* $\qquad \|x\|_\infty = \max_k(|x_k|)$ *are convex.*
- *The TV norm definite as in 2.1.11 of* $u : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}$

  $$TV(u) = \int_\Omega |\nabla u| \, dxdy$$

  is a convex functional.

Definition 2.1.13 (**Open Set**). *For a given normed space* V *a subset* $A \subset V$ *is said to be open if for each point* $u \in A$ *there exists* $\delta > 0$ *such that* $\|u - v\| < \delta$ *for all* $v \in A$

Definition 2.1.14 *If $A \in \mathbf{V}$ then $C(A) = \mathbf{V} \backslash A$ denotes the complement of the set A in $\mathbf{V}$, that is, the set of all points $x \in A$ which do not belong to A.*

Definition 2.1.15 (**Closed Set**). *A subset $A \in \mathbf{V}$ is closed if its complement $C(A)$ is open.*

Definition 2.1.16 (**Lipschitz Condition**). *If for any points $x, y \in S \subset \mathbb{R}$ for some $M \in \mathbb{R}$ the real function $f : S \to \mathbb{R}$ satisfies*

$$|f(x) - f(y)| \leq M|x - y|$$

then f is said to satisfy the Lipschitz condition in S and is called a Lipschitz continuous function.

Theorem 2.1.17 (Young's Inequality for Products). *If a and b are nonnegative real numbers and p and q are positive real numbers such that $\frac{1}{q} + \frac{1}{p} = 1$, then*

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}.$$

Equality holds if and only if $a^p = b^q$.

Theorem 2.1.18 (Hölder's Inequality). *Let $\Omega \in \mathbb{R}^n$ be a domain and $f \in L^p(\Omega), g \in L^q(\Omega)$ with $1 \leq p, q \leq \infty$ such that $\frac{1}{q} + \frac{1}{p} = 1$. Then $fg \in L^1(\Omega)$ and $\|fg\|_{L^1(\Omega)} \leq \|f\|_{L^p(\Omega)} \|g\|_{Lq(\Omega)}$.*

Theorem 2.1.19 (Minkowski's Inequality). *Let Ω be a normed space and $1 \leq p \leq \infty$ and $f, g \in L^p(\Omega)$. Then $f + g \in L^p(\Omega)$ and*

$$\|f + g\|_{L^p(\Omega)} \leq \|f\|_{L^p(\Omega)} + \|g\|_{L^p(\Omega)}.$$

The Minkowski inequality establishes that for the L^p spaces the triangle inequality is satisfied and that the set of p^{th} power integrable functions, together with the function $\|\cdot\|_{L^p}$, is a normed vector space.

2.2 Curves, Surfaces and Some Calculus and Geometry Elements

2.2.1 Curves and Surfaces in Euclidean Spaces

Considering the space $\mathbb{R}^2$, the lower-dimensional interface is a curve that separates $\mathbb{R}^2$ into subdomains with nonzero areas. Generally speaking, a curve is an object similar to a line but which is not required to be straight. In this thesis we are limiting our interface curves to those that are closed in $\mathbb{R}^2$ or a subdomain $\Omega \subseteq \mathbb{R}^2$, and we denote the interface by $\partial\Omega$. In other words the curves we consider have clearly defined interior and exterior regions.

Example 2.2.1 *For $\mathbf{x} = (x, y) \in \mathbb{R}^2$ consider $\phi(\mathbf{x}) = x^2 + y^2 - 1$. The interface is defined by $\phi(\mathbf{x}) = 0$, or alternatively, as the isocontour of the unit circle function*

defined by $\partial\Omega = \{\mathbf{x} \ : \ |\mathbf{x}| \leq 1\}$. *We also define the interior region, which is a unit open disk* $\Omega^- = \{\mathbf{x} \ : \ |\mathbf{x}| < 1\}$, *and the exterior region* $\Omega^+ = \{\mathbf{x} \ : \ |\mathbf{x}| > 1\}$. *These regions are depicted in Fig. 2.1.*

Note that a line is a special case of curve, a curve with null curvature[1].

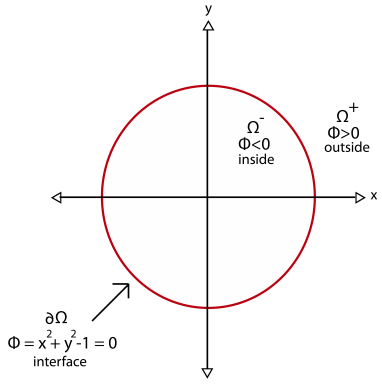


Figure 2.1: Representation of the curve $x^2 + y^2 = 1$.

The curve described above is given in an analytical way. In general, one needs to parameterize the curve with a vector function $\phi = (x(t), y(t))$, where the parameter t is given in an interval $[t_a, t_b]$, such that the condition of being a closed curve implies, $\mathbf{x}(t_a) = \mathbf{x}(t_b)$. The parametric equation for the circle above would be

$$\phi(t) = (\cos(t), \sin(t)) \quad \text{for } 0 \leq t < 2\pi.$$

In the case of three spatial dimensions the lower-dimensional interface is a surface that separates $\mathbb{R}^3$ into separate subdomains with nonzero volumes. A surfaces is a two-dimensional topological manifold whith clearly defined interior and exterior regions.

Example 2.2.2 *For* $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ *consider* $\phi(\mathbf{x}) = x^2 + y^2 + z^2 - 1$. *The interface is defined by* $\phi(\mathbf{x}) = 0$, *or alternatively, the boundary of the unit sphere is defined by* $\partial\Omega = \{\mathbf{x} \ : \ |\mathbf{x}| \leq 1\}$. *The interior region can be defined as an open unit sphere* $\Omega^- = \{\mathbf{x} \ : \ |\mathbf{x}| < 1\}$, *and the exterior region* $\Omega^+ = \{\mathbf{x} \ : \ |\mathbf{x}| > 1\}$.

2.2.2 Gradient, Mean Curvature and Some Geometry Element

Here we will give some definitions of geometric characteristics of the interface, starting with the definition of the gradient.

Definition 2.2.3 *For a given scalar function* $\phi(x_1, x_1, ..., x_n)$ *the gradient is denoted as* $\nabla\phi$ *or grad* ϕ *and is defined as*

$$\nabla\phi = \left(\frac{\partial\phi}{\partial x_1}, \frac{\partial\phi}{\partial x_2}, ..., \frac{\partial\phi}{\partial x_n} \right).$$

[1] the definition of a curvature is given by definition 2.6

The gradient $\nabla\phi$ is perpendicular to the isocontours of ϕ and points in the direction of increasing ϕ. The unit (outward) normal vector $\mathbf{n}$ is a vector that points in the same direction as the gradient $\nabla\phi$ for points on the interface, and is defined as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}.$$

Definition 2.2.4 *The mean curvature, or simply curvature, of the interface is defined as the divergence of the unit normal* $\mathbf{n}$

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} = \frac{\partial}{\partial x_1}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \frac{\partial}{\partial x_2}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + ... + \frac{\partial}{\partial x_n}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \quad (2.6)$$

It can be shown that $\kappa > 0$ for convex regions, $\kappa < 0$ for concave regions, and $\kappa = 0$ for a plane; shown in Fig. 2.2. In two dimensional space the curvature is equal to

$$\kappa = \frac{\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx}}{(\phi_x^2 + \phi_y^2)^{3/2}},$$

and in three dimensional space

$$\kappa = \frac{\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx} + \phi_x^2\phi_{zz} - 2\phi_x\phi_z\phi_{xz} + \phi_z^2\phi_{xx} + \phi_y^2\phi_{zz} - 2\phi_y\phi_z\phi_{yz} + \phi_z^2\phi_{yy}}{|\nabla\phi|^3}.$$

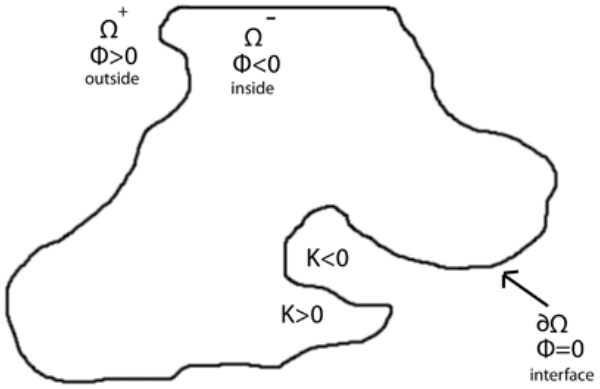


Figure 2.2: Representation of the curvature with convex regions $\kappa > 0$, and concave regions $\kappa < 0$.

2.2.3 Heaviside and Dirac Delta Functions

A functions whose values change abruptly by switching on or off and showing a specified value of ϕ on the point $\mathbf{x}$ is described mathematically by the function called the Unit Step Function (otherwise known as the Heaviside function). This function helps us to deal with functions that are piecewise continuous, i.e., have finite jump discontinuities at finitely many places. The definition of the Heaviside function is well known and is

as follows:

Definition 2.2.5 *The Heaviside function, usually denoted by H, is a non-continuous function whose value is zero for negative arguments and one for positive argument. For a given function $\phi(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$ the Heaviside function is written:*

$$H(\phi) = \begin{cases} 1 & \text{if} \quad \phi \geq 0 \\ 0 & \text{if} \quad \phi < 0 \end{cases}$$

Note that the characteristic function of the interior and exterior regions, denoted by χ^+ and χ^- respectively, can be expressed in terms of the Heaviside function as follows

$$\chi^- = H(\phi) \text{ and } \chi^+ = 1 - H(\phi)$$

for all $\mathbf{x}$.

Definition 2.2.6 *For a given function $\phi(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, the directional derivative of the Heaviside function H in the normal direction $\mathbf{n}$ is called Dirac delta function and is equal to*

$$\widehat{\delta}(\mathbf{x}) = H(\phi(\mathbf{x}))' \cdot \mathbf{n} = H'(\phi(\mathbf{x})) \nabla(\phi(\mathbf{x})) \cdot \frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|} = H'(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})| \tag{2.7}$$

In one dimensional space, the delta function is defined as the derivative of the Heaviside function, $\delta(\phi) = H'(\phi)$. The delta function $\delta(\phi)$ is identically zero everywhere except at $\phi = 0$. This allows us to use 1-dimensional Dirac delta functions to rewrite the n-dimensional Dirac delta function given by equation (2.7) as

$$\widehat{\delta}(\mathbf{x}) = \delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|. \tag{2.8}$$

For a given function $f(\mathbf{x})$ defined on Ω, the volume integral inside the boundary $\Gamma = \partial\Omega$ is the integral of $f(\mathbf{x})$ over the exterior region χ^-

$$\int_\Omega f(\mathbf{x})H(\phi)d\mathbf{x}.$$

In similar way,

$$\int_\Omega f(\mathbf{x})(1 - H(\phi))d\mathbf{x}$$

representing the integral of $f(\mathbf{x})$ over the interior region χ^+. The surface integral of a function $f(\mathbf{x})$ over a boundary Γ is defined as

$$\int_\Omega f(\mathbf{x})|\nabla H(\phi)|d\mathbf{x} = \int_\Omega f(\mathbf{x})\delta(\phi)|\nabla\phi|d\mathbf{x}.$$

Note that if $f(\mathbf{x}) = 1$, then this yields the surface area or volume of $\partial\Omega$. In this way, for example in $\mathbb{R}^2$, the area formula above can be rewritten

$$\int_{\mathbb{R}^2} H(\phi)dxdy$$

and the length of the interface Γ is

$$\int_{\mathbb{R}^2} |\nabla H(\phi)| dxdy = \int_{\mathbb{R}^2} \delta(\phi)|\nabla\phi| dxdy.$$

2.3 Calculus of Variation

Calculus of variations seeks to find the optimal shape, curve, surface, or processes when the optimality criterion is given in form of the integral of an unknown function. Mathematically, this involves finding an appropriate function that makes a given quantity (usually an energy or integral) stationary which, in physical problems, is usually a minimum or maximum. Because a function is varied, these problems are called variational and solved by the so-called calculus of variations. Calculus of variations highlights the interactions between analysts, geometers, and physicists and its reach in application problems which are fundamental in many areas of mathematics, physics, engineering and other applications. A good impression of this diversity can be obtained by reading the book entitled "The Parsimonious Universe"[74].

In this section we introduce the basic tools to compute the first variation (also known as the Euler-Lagrange equation) of a functional. Extensive literature in this respect can be found in the monographies [58, 60, 61, 138, 128] or elsewhere.

2.3.1 Variation of a Functional

A functional is a function of another function (such as a curve, or a surface, etc.) which assigns a real number to each function in some class. The *first variation* of a functional deals with the problem of finding a function for which the value of a certain integral is either the largest or the smallest possible and the integrands of which are functions of independent variables, dependent variables, and the derivatives of one or more dependent variables. Classical solutions to minimization problems in the calculus of variations are prescribed by boundary value problems involving certain types of differential equations, known as Euler–Lagrange equations.

Consider the general functional $\mathfrak{J}(u) : \Omega \to \mathbb{R}$

$$\mathfrak{J}(u) = \int_\Omega F(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}))\, \mathbf{dx}, \tag{2.9}$$

where Ω denotes some normed linear space (for example, Ω=$\mathbb{R}^n$, $n \geq 1$) which is a solution space of the function u, $\nabla u(\mathbf{x})$ denotes its gradient $\nabla u(\mathbf{x}) = (u(\mathbf{x})_{x_1}, u(\mathbf{x})_{x_2}, \ldots, u(\mathbf{x})_{x_n})$. Here $\mathbf{dx}$ is the n-differential element defined as $\mathbf{dx} = dx_1 dx_2 \cdots dx_n$.

Example 2.3.1 *Simple examples of variational integrals (2.9) are the Dirichlet integral*

$$\mathfrak{D}(u) = \frac{1}{2} \int_\Omega |\nabla u(\mathbf{x})|^2\, \mathbf{dx}, \tag{2.10}$$

and the nonparametric arc length integral

$$\mathfrak{L}(u) = \int_{\Omega} \sqrt{1 + |\nabla u(\mathbf{x})|^2}\, \mathbf{dx}. \tag{2.11}$$

The minimisation problem, in other words the first variation, consists of solving the following minimization problem:

$$\min_{u} \mathfrak{J}(u). \tag{2.12}$$

The most important *necessary* condition to be satisfied by any minimizer of a variational integral $\mathfrak{J}(u)$ is the vanishing of its first variation $\delta\mathfrak{J}(u)$

$$\delta\mathfrak{J}(u) = \frac{d}{d\varepsilon}\mathfrak{J}(u + \varepsilon\varphi)\Big|_{\varepsilon=0} = 0, \tag{2.13}$$

where $\varphi \in \Omega$ is a test function and ε is a real parameter (which is restricted to some interval around 0). For some $u_0 \in \Omega$, we call $\delta\mathfrak{J}(u_0)$ the first variation of $\mathfrak{J}$ at u_0 in the direction of φ.

Example 2.3.2 *A concrete example of both mathematical and practical importance would be the minimal curve problem of finding the shortest path between two specified locations. Given two distinct points (a, α) and (b, β) in $\mathbb{R}^2$ the task is to find the curve of*

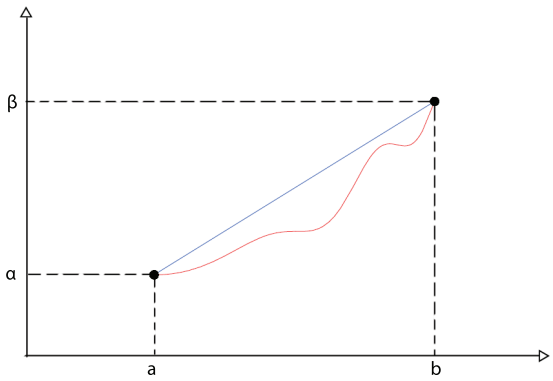


Figure 2.3: The shortest path is a straight line.

shortest length connecting them. We know by intuition that the shortest route between two points is a straight line; see Fig.2.3, with equation of the form

$$y = cx + d = \frac{\beta - \alpha}{b - a}(x - a) - \alpha. \tag{2.14}$$

In terms of calculus of variation we might formulate the minimal curve problem in a mathematically precise way. Let us assume that the minimal curve is given as the graph of a smooth continuous function $y = u(x)$ with a piecewise continuous derivative ($y \in C^1$). According to the length of the curve given by (2.11), with $\nabla u(x) = u'(x) = du/dx$, the minimization function that satisfies the boundary condition $u(a) = \alpha$ and $u(b) = \beta$

would be written

$$\min_u \int_a^b \sqrt{1+|u'(x)|^2}dx. \tag{2.15}$$

The minimal curve problem asks us to find the function $y = u(x)$ that minimizes the arc length functional (2.11) among all functions satisfying the prescribed boundary conditions. Obviously we can notice that the function (2.14) is the one that minimizes the arc length integral (2.15) subject to the given boundary conditions, which proves that our intuition is indeed correct.

Another example is to construct the geodesics on a curved surface, meaning the curves of minimal length connecting two given points a, b lying on a surface $S \in \mathbb{R}^3$, or the minimal surface problem as a natural generalization of the minimal curve or geodesic problem.

2.3.2 Gâteaux Derivative of a Functional

Definition 2.3.3 (**Gâteux derivative**). *Let $\mathfrak{J}$ be a function on an open subset U of a Banach space V ($U \subseteq V$) in the Banach space Y. Then $\mathfrak{J}$ is called Gâteux differentiable for $u \in U$ in the direction of $\varphi \in V$ if the first directional derivative $\mathfrak{J}'(u;\varphi)$ exists for each test function $\varphi \in V$; that is, if*

$$\delta\mathfrak{J}(u;\varphi) = \delta\mathfrak{J}|_u(\varphi) = \lim_{\varepsilon\to 0}\frac{J(u+\varepsilon\varphi)-J(u)}{\varepsilon}. \tag{2.16}$$

The first variation as defined above corresponds to the Gâteaux derivative of $\mathfrak{J}(u)$, which is just the usual derivative of $\mathfrak{J}(u+\varepsilon\varphi)$ with respect to ε (for fixed u and φ) evaluated at $\varepsilon = 0$.

Example 2.3.4 (**The exponential function in $\mathbb{R}^+$.**) *Let us consider the exponential function $f(u) = e^u : \mathbb{R} \to \mathbb{R}^+$ which can be written as the infinite series*

$$e^u = 1 + u + \frac{u^2}{2} + \ldots + \frac{u^k}{k!} + \ldots.$$

By straightforward calculation, given $e^{a+b} = e^a e^b$ and the series defining e^x, we can compute the Gâteux derivative

$$\delta(e^u;\varphi) = \lim_{\varepsilon\to 0}\frac{e^u e^{\varepsilon\varphi}-e^u}{\varepsilon} = e^u \lim_{\varepsilon\to 0}\frac{e^{\varepsilon\varphi}-1}{\varepsilon} = \varphi e^u.$$

Example 2.3.5 (**The absolute value function in $\mathbb{R}$.**) *Let $f(u) = |u| : \mathbb{R} \to \mathbb{R}^+$. Using the calculation of the limit in the Gâteux derivative we have*

$$\delta(u;\varphi) = \begin{cases} \varphi\frac{u}{|u|} & u \neq 0, \\ \varphi & u = 0. \end{cases}$$

Notice that the Gâteux differentials of $|u|$ do exist at zero; however, at zero, the Gâteux differentials depend on φ in a nonlinear way.

Definition 2.3.6 (**Local Minimiser**). *A real-valued functional $\mathfrak{J} : U \to \mathbb{R}$, defined on a subset U of the normed space V, is said to have a local minimiser at the point $\tilde{u}$,*

relative to the norm $\|\cdot\|$ *if there exists some* $\epsilon > 0$ *such that*

$$\mathfrak{J}(\tilde{u}) \leq \mathfrak{J}(u) \ \forall u \in B_\epsilon(\tilde{u}) \bigcap U; \tag{2.17}$$

with $B_\epsilon(\tilde{u}) := \{u \in V : \|u - \tilde{u}\| < \epsilon\}$.

In the same way the Local Maximiser can be defined by replacing the inequality 2.17 with $\mathfrak{J}(\tilde{u}) \geq \mathfrak{J}(u)$. The value of the function at this point is called minimum (maximum) of the function.

Definition 2.3.7 (**Global Minimiser**). *A real-valued functional* $\mathfrak{J} : U \to \mathbb{R}$*, is said to have a global minimiser at the point* $\tilde{u}$*, if* $\mathfrak{J}(\tilde{u}) \leq \mathfrak{J}(u)$ *for all* u.

In the same way we get the Glomal Maximiser if we have instead the inequality $\mathfrak{J}(\tilde{u}) \geq \mathfrak{J}(u)$. The global minimum and global maximum points are also known as the *arg min* and *arg max*: the argument (input) at which the minimum (respectively, maximum) occurs.

Definition 2.3.8 (**Stationary Point**). *Let* $\mathfrak{J} : U \to \mathbb{R}$ *be a function with solution space* $U \subset V$*. Suppose that for some* $\tilde{u} \in U$*,* $\mathfrak{J}$ *is Gâteaux-differentiable for all test functions* $\varphi \in V$*. Then* $\tilde{u} \in U$ *is said to be a stationary point of* $\mathfrak{J}$ *if* $\delta\mathfrak{J}(\tilde{u}; \varphi) = 0$ *for all* $\varphi \in V$.

If the Gâteux derivative exists at a point $u \in U$ and if the problem $\min_{u \in U} \mathfrak{J}(u)$ has a solution u, then we have $\delta\mathfrak{J}(u, v) = 0$. Conversely, if $\mathfrak{J}$ is convex, then a solution u of $\delta\mathfrak{J}(u, v) = 0, \forall v \in U$ is a solution of the minimization problem. The equation $\delta\mathfrak{J}(u, v) = 0$ is called an Euler-Lagrange equation of the minimization problem $\min_{u \in U} \delta\mathfrak{J}(u)$.

Theorem 2.3.9 (**Necessary Condition for a Local Minimiser**). *For a given Gâteux-differentiable function* $\mathfrak{J} : U \to \mathbb{R}$*, if* $\tilde{u}$ *is a local minimiser of* $\mathfrak{J}(u)$*, then* $\tilde{u}$ *is a stationary point of* $\mathfrak{J}(u)$.

The proof of this theorem can be found in [11, 49, 86].

2.3.3 The Gauss (Divergence) Theorem

The divergence theorem, known in literature as Gauss's theorem or also known as the Gauss-Ostrogradsky theorem, is a theorem in vector calculus that can be stated as follows.

Theorem 2.3.10 (**The Gauss Theorem**). *Let* $\mathbf{F}$ *be a continuously differentiable vector field in a domain* $V \subset \mathbb{R}^n$*. Let* $\Omega \subset V$ *be a closed, bounded region whose boundary is a smooth surface,* $\partial\Omega$*. Then the volume integral of the divergence of* F *(denoted* $divF$ *or* $\nabla \cdot F$*) over* Ω *and the surface integral of* F *over the boundary* $\partial\Omega$ *of* Ω *are related by*

$$\int_\Omega (\nabla \cdot \mathbf{F})\, \mathbf{dx} = \int_{\partial\mathbf{\Omega}} \mathbf{F} \cdot \mathbf{n}\, \mathbf{ds}, \tag{2.18}$$

where $\nabla \cdot \mathbf{F} = \frac{\partial \mathbf{F}}{x_1} + \frac{\partial \mathbf{F}}{x_2} + \ldots + \frac{\partial \mathbf{F}}{x_n}$*,* $d\mathbf{x} = \{dx_1, dx_2 \ldots dx_n\}$*, ds indicating integration with respect to surface area on* $\partial\Omega$*, and* $\mathbf{n}$ *the unit outward for each point* $\mathbf{x} \in \partial\Omega$*, normal to* $\partial\Omega$.

2.3.4 Integration by Parts

A special rule, as an immediate consequence of the above divergence theorem, is the integration by parts formula, available for integrating products of two functions. By applying (2.18) to the product of a scalar function $\mathbf{g}$ and a vector field $\mathbf{F}$ we obtain the vectorial representation

$$\int_{\Omega}(\mathbf{F}\cdot\nabla\mathbf{g}+\mathbf{g}\nabla\cdot\mathbf{F})\,\mathbf{dx}=\int_{\partial\mathbf{\Omega}}\mathbf{gF}\cdot\mathbf{n}\,\mathbf{ds}. \tag{2.19}$$

In the 1-dimensional case, for $\mathbf{F}=u(x)$, $\mathbf{g}=v(x)$, and the differentials $\nabla\mathbf{F}=u'(x)dx$ and $\nabla\mathbf{g}=v'(x)dx$, then integration by parts can be written more compactly

$$\int u(x)v'(x)\,dx=u(x)v(x)-\int u'(x)v(x)\,dx.$$

Example 2.3.11 *Consider the problem of finding the first variation* $\frac{d}{d\varepsilon}\mathfrak{J}(u+\varepsilon\varphi)\big|_{\varepsilon=0}=0$ *of the functional*

$$\mathfrak{J}(u)=\int_{\Omega}|\nabla u|\,dxdy,$$

defined on a domain $\Omega\subset\mathbb{R}^2$*. Recall that* $\varepsilon\varphi$ *is composed of the parameter* $\varepsilon\to 0$ *and the continuously differentiable test function* φ *in* Ω*. Then we compute,*

$$\frac{d}{d\varepsilon}\mathfrak{J}(u+\varepsilon\varphi)\bigg|_{\varepsilon=0}=\frac{d}{d\varepsilon}\int_{\Omega}|\nabla(u+\varepsilon\varphi)|\,dxdy\bigg|_{\varepsilon=0}=\int_{\Omega}\frac{\nabla(u+\varepsilon\varphi)}{|\nabla(u+\varepsilon\varphi)|}\cdot\nabla\varphi\,dxdy\bigg|_{\varepsilon=0}=\int_{\Omega}\frac{\nabla u}{|\nabla u|}\cdot\nabla\varphi\,dxdy.$$

Using integration by parts on the above, we get

$$\int_{\Omega}\frac{\nabla u}{|\nabla u|}\cdot\nabla\varphi\,dxdy=\int_{\partial\Omega}\mathbf{n}\cdot\frac{\nabla u}{|\nabla u|}\,\mathbf{ds}-\int_{\Omega}\nabla\cdot\left(\frac{\nabla u}{|\nabla u|}\right)\varphi\,dxdy, \tag{2.20}$$

where $\partial\Omega$ *is the boundary of* Ω*,* $\mathbf{n}$ *is the unit outward normal vector to* $\partial\Omega$ *and* $\mathbf{ds}$ *is the length element of integration. Requiring*

$$\frac{d}{d\varepsilon}\mathfrak{J}(u+\varepsilon\varphi)\bigg|_{\varepsilon=0}=0,$$

for all test functions φ *then the following partial differential equation must be satisfied:*

$$\nabla\cdot\left(\frac{\nabla u}{|\nabla u|}\right)=0\ \ in\ \ \Omega, \tag{2.21}$$

with Neumann boundary condition $\mathbf{n}\cdot\nabla\mathbf{u}=\mathbf{0}$ *on* $\partial\Omega$ *. The above equation is known as the Euler-Lagrange equation.*

2.4 Bounded Variation and Related Properties

In this subsection we introduce the notation of functions of bounded variation (BV) which will be used in the later chapters. More details can be found in [187, 63, 89].

Definition 2.4.1 (**Compact Set**). *A subset* $\mathbf{U}$ *of a metric space* $(\mathbf{V}, \|\cdot\|)$ *is said to be compact if for every arbitrary collection* $\{U_i\}_{i\in I}$ *of subsets of* $\mathbf{U}$ *such that* $\mathbf{U} \subset \bigcup_{i\in I} U_i$, *there is a finite subset* J *of* I *such that* $\mathbf{U} \subset \bigcup_{j\in J} U_j$.

Definition 2.4.2 (**Support of a Function**). *The support of a function* f *is the set of points where this function is not zero* $\{x : f(x) \neq 0\}$, *or the closure of that set.*

It means that the support of the function is a compact set, or equivalently, contained in a finite closed interval.

Definition 2.4.3 (**Compact Support**). *Functions with compact support in a space* $\mathbf{V}$ *are those for which their support is a compact subset of* $\mathbf{V}$.

A function has compact support if it is zero outside of a compact set.

Definition 2.4.4 (**Total Variation (TV) Seminorm**). *Let* Ω *be a bounded open subset of* $\mathbb{R}^n$ *and let* u *be a function in* $L^1(\Omega)$. *Define the total variation,*

$$TV(u) = \|Du\|(\Omega) = \int_\Omega |Du|(\boldsymbol{x})d\boldsymbol{x} \ = \sup_V \left\{ \int_\Omega u(\boldsymbol{x}) \ div\, \varphi(\boldsymbol{x}) \ d\boldsymbol{x} \right\}, \tag{2.22}$$

where V *is the set of test functions*

$$V = \{ \ \boldsymbol{\varphi} = (\varphi_1, \varphi_2 \ldots, \varphi_n) \in C_0^1(\Omega; \mathbb{R}^n) : \ |\varphi(\boldsymbol{x})|_{L^\infty(\Omega)} \leq 1, \ \forall \boldsymbol{x} \in \Omega\}, \tag{2.23}$$

$div\, \varphi(\boldsymbol{x}) = \nabla \cdot \boldsymbol{\varphi} = \sum_{i=1}^{n} \frac{\partial \varphi}{\partial x_i}(\boldsymbol{x})$, $\boldsymbol{dx}$ *is the Lebesgue measure*[2] *and* $\boldsymbol{\varphi}$ *is a vector valued function with compact support. Here* $C_0^1(\Omega)$ *is the space of continuously differentiable functions with compact support in* Ω.

A particular and interesting case is when $u \in C^1(\Omega)$, then integration by parts gives

$$\int_\Omega u \ \mathrm{div}\boldsymbol{\varphi} \ \boldsymbol{dx} = -\int_\Omega \sum_{i=1}^{n} \frac{\partial u}{\partial x_i} \varphi_i \ \boldsymbol{dx}, \tag{2.24}$$

for every $\boldsymbol{\varphi} \in C_0^1(\Omega; \mathbb{R}^n)$, so that

$$\int_\Omega |Du| = \int_\Omega |\nabla u| \ \boldsymbol{dx}, \tag{2.25}$$

where $\nabla u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \ldots, \frac{\partial u}{\partial x_n} \right)$.

Definition 2.4.5 (**Bounded Variation (BV)**). *If for a function* $u \in L^1(\Omega)$, $\|Du\|(\Omega) < \infty$, *then the function* u *is known as one of bounded variation.*

The notation $\mathrm{BV}(\Omega)$ denotes all functions in $L^1(\Omega)$ that are of bounded variation.

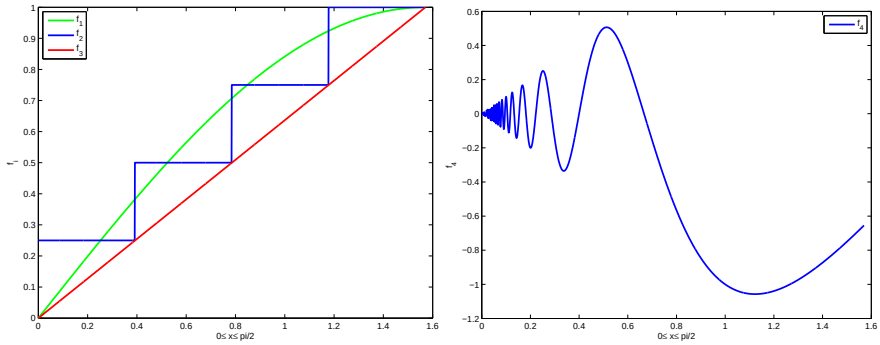
[2] In Euclidean spaces, the standard way to assign a *measure* (length, area or volume) to a given subset is through the Lebesgue measure. Hence, sets with finite Lebesgue measure are called Lebesgue measurables. In real analysis, this measure is used to define Lebesgue integration.

Definition 2.4.6 *A function $u : [a, b] \to \mathbb{R}$ is said to be of bounded variation on $[a, b]$ if and only if there exists a constant $K > 0$ such that*

$$\sum_{i=1}^{n} |u(x_i) - u(x_{i-1})| \leq K$$

for all partitions $P = \{x_0, x_1, \ldots, x_n\}$ of $[a, b]$.

To explain it more we give the one-dimensional definition of BV and give some illustrative examples.



(a) three bounded variation functions (b) a function of no bounded variation with the same total variation

Figure 2.4: Illustration of bounded and non bounded variation functions.

Example 2.4.7 *The following functions $f_1(x), f_2(x)$ and $f_3(x)$ given by*

$$f_1(x) = \sin(x), \tag{2.26}$$

$$f_2(x) = \begin{cases} 1/4 & \text{for } 0 \leq x < \pi/8 \\ 1/2 & \text{for } \pi/8 \leq x < \pi/4 \\ 3/4 & \text{for } \pi/4 \leq x < 3\pi/8 \\ 1 & \text{for } 3\pi/8 \leq x < \pi/2 \end{cases}, \tag{2.27}$$

$$f_3(x) = \frac{2x}{\pi}, \tag{2.28}$$

belong to $BV(\Omega)$ with $\Omega = \{x \in [0, \pi/2]\}$. Moreover, it is not difficult to see that all of the above functions have the same total variation equal to one, Fig. 2.4 (a).

Example 2.4.8 *Now consider the function $f_4(x)$ defined as*

$$f_4(x) = \begin{cases} 0 & \text{for } x = 0 \\ x\cos(\pi/x) & \text{for } 0 < x \leq a \ \text{ with } \ a > 0. \end{cases} \tag{2.29}$$

Here $\Omega = \{x \in [0, a]\}$ for any $a > 0$, Fig. 2.4 (b). We see that as $x \to 0$ the frequency of the oscillations of $f_4(x)$ increases, then the closer x gets to zero the more variations need to be added and the value of the integral (2.25) increases. Therefore, this function has infinite total variation and does not belong to $BV(\Omega)$.

Remark 2.4.9

- *Under the norm* $\|u\|_{BV} = \|u\|_{L^1} + \int_\Omega |Du|$, $BV(\Omega)$ *is a Banach space.*
- *The concept of the function with bounded variation maked sense only on compact intervals.*
- *A Lipschitz function is with bounded variation.*

An interesting characterization of the functions with total variation comes from the following example.

Example 2.4.10 *Let* χ_E *be a characteristic (or indicator) function of a set* E

$$\chi_E = \begin{cases} 1 & if\, \mathbf{x} \in E \\ 0 & if\, \mathbf{x} \in \Omega - E \end{cases}.$$

Computing the total variation, one maximizes over all vector fields $\varphi \in C^1(\Omega, \mathbb{R}^n)$*;* $\|\varphi\|_\infty \leq 1$*:*

$$\int_\Omega \chi_E \text{div}\, \varphi \, \mathbf{dx} = \int_E \text{div}\, \varphi \, \mathbf{dx} = \int_{\partial E} \mathbf{n} \cdot \varphi \, \mathbf{ds} \text{ (Gauss}' \text{theorem)},$$

where $\mathbf{n}$ *is the outward unit normal to* ∂E*. The expression is maximized for any vector field with* $\varphi|_{\partial E} = \mathbf{n}$*, hence*

$$TV(\chi_E) = \int_{\partial E} \mathbf{ds} = \mathcal{H}_{n-1}.$$

Here, $\mathcal{H}_{n-1}$*, is the* $(n-1)$*-dimensional Haussdorff measure, i.e. the length in the case* $n = 2$*, or area for* $n = 3$*.*

This result shows that the total variation of a given image u is just the sum of every length of all its λ-level curves. In this way all the discontinuities of u and therefore the contribution of the edges to the total variation integral is enforced.

2.4.1 Co-area Formula

The co-area formula gives a natural connection between the total variation of a function $u(\mathbf{x})$ representing an image and the perimeter of its level sets. For an open set in Euclidean space, the co-area formula states that the total variation of a function can be computed by integrating the lengths of its level lines. In other words, for a real-valued Lipschitz function $u(\mathbf{x})$ in an open set Ω in R^n, the total variation of $u(\mathbf{x})$ can be computed by summing the perimeters of all level sets of $u(\mathbf{x})$.

More precisely, for a function $u(\mathbf{x}) \in BV(\Omega)$ defined in Ω, an open set in $\mathbb{R}^n$, we define the level domain, a smooth $(n-1)$-dimensional hyper-surface in $\mathbb{R}^n$, as

$$E_\lambda = \{\mathbf{x} \in \Omega : u(\mathbf{x}) = \lambda\}.$$

Fig. 2.5 illustrates this with an example. Then for any continuous and integrable function $g(\mathbf{x}) \in C^1$, we have

$$\int_{\mathbb{R}^n} g(\mathbf{x})\, |\nabla u(\mathbf{x})|\, \mathbf{dx} = \int_{-\infty}^{+\infty} \left(\int_{E_\lambda} g(\mathbf{x})\, ds \right)\, d\lambda.$$

For the particular case when $g(\mathbf{x}) = 1$ and the region of integration is a subset $\Omega \subset \mathbb{R}^n$ we have

$$\int_{\Omega} |\nabla u|\, \mathbf{dx} = \int_{-\infty}^{+\infty} \left(\int_{E_\lambda} ds \right)\, d\lambda = \int_{-\infty}^{\infty} d\lambda \int_{\Omega} |D\chi_{E_\lambda}|\, d\mathbf{x}.$$

Definition 2.4.11 (**Perimeter**). *Let E_λ be a cumulative level set in Ω, an open set in $\mathbb{R}^n$. Define the perimeter of E_λ in Ω as*

$$Per(E_\lambda) = \int_{\Omega} |D\chi_{E_\lambda}| d\mathbf{x} = sup \left\{ \int_{E_\lambda} \text{div}\varphi\ \mathbf{dx} : \varphi \in C_0^1(\Omega; \mathbb{R}^n)\ and\ |\varphi(\mathbf{x})| \leq 1 \right\}.$$

Definition 2.4.12 (**Coarea Formula**). *Assume that u is Lipschitz continuous and that for almost every $\lambda \in \mathbb{R}$, the level set is a smooth $(n-1)$-dimensional hyper-surface in $\mathbb{R}^n$. Then*

$$\|Du\| = \int_{\Omega} |Du| d\mathbf{x} = \int_{-\infty}^{\infty} Per(E_\lambda) d\lambda. \tag{2.30}$$

The proof can be found in [63].

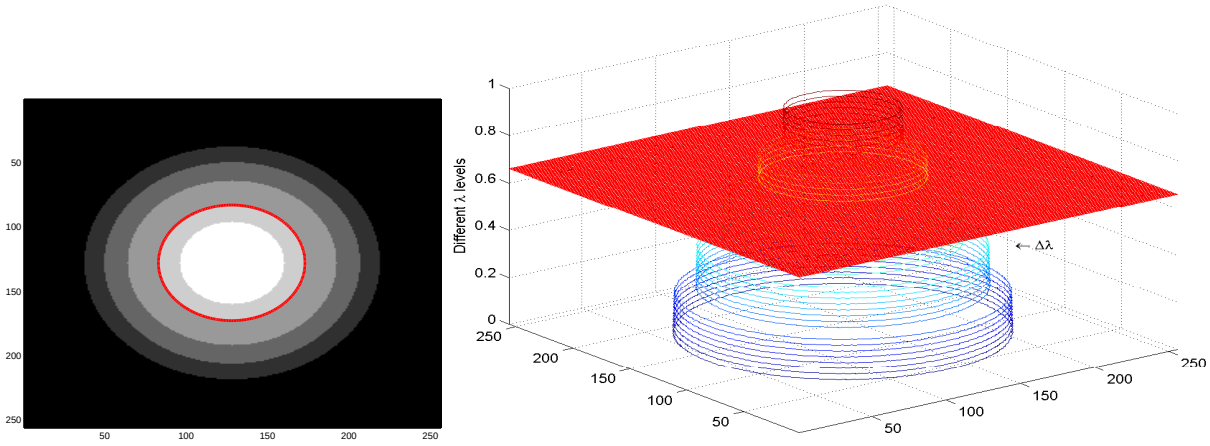


Figure 2.5: On the left is a given gray level image $u(\mathbf{x})$ and on the right some of its λ-level curves, where $u(\mathbf{x}) = \lambda$ for some $\lambda \in [0, 1]$.

2.5 Discrete PDEs and Notation

There are situations when analytically solving a PDE can be hard or even impossible, or when the data is only known at a certain number of discrete locations. In these situations one has to solve a discrete version of a continuous PDE. In general, a continuous linear boundary value problem in d-dimensions is denoted by:

$$L^{\Omega} u(\mathbf{x}) = f^{\Omega}(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Omega, \tag{2.31}$$

$$L^{\Gamma}u(\mathbf{x}) = f^{\Gamma}(\mathbf{x}) \text{ for } \mathbf{x} \in \Gamma, \tag{2.32}$$

where Ω is a bounded open domain in $\mathbb{R}^d$, $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and Γ is its boundary.

Example 2.5.1 *One of the most common examples would be the classical Poisson's equation in a two-dimensional problem with Dirichlet boundary conditions, given by*

$$\Delta^{\Omega}u(\mathbf{x}) = f^{\Omega}(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Omega, \tag{2.33}$$

$$u(\mathbf{x}) = f^{\Gamma}(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Gamma, \tag{2.34}$$

Similarly, a continuous nonlinear boundary value problem is defined by:

$$N^{\Omega}u(\mathbf{x}) = f^{\Omega}(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Omega \tag{2.35}$$

$$N^{\Gamma}u(\mathbf{x}) = f^{\Gamma}(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Gamma, \tag{2.36}$$

where N is a nonlinear operator.

Example 2.5.2 *An example of resulting nonlinear boundary value problems which will be seen in later chapters is the well known variational image denoising model by Rudin, Osher and Fatemi [135], given as follows:*

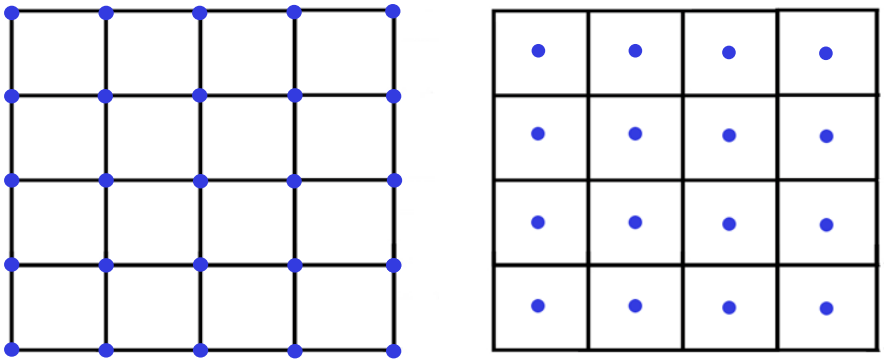
$$-\nabla \cdot \Big(\frac{\nabla u(\mathbf{x})}{|\nabla u(\mathbf{x})|}\Big) + \lambda(u(\mathbf{x}) - u^0(\mathbf{x})) = 0 \ \text{ in } \ \Omega, \tag{2.37}$$

$$\frac{\partial u(\mathbf{x})}{\partial n} = 0 \text{ in } \partial\Omega \tag{2.38}$$

where $\lambda > 0$ is a constant, $u^0(\mathbf{x})$ is the noisy image and $u(\mathbf{x})$ is the true image which we wish to recover.

In this thesis we deal with image domains which are usually rectangular and where the values of f are known uniformly distributed points in the domain $\Omega \subset \mathbb{R}^d$. Therefore, a natural choice for discretising the domain is to use the finite difference method. We will restrict our discussion to $\Omega = (a, b) \times (c, d) \subset \mathbb{R}^2$, which turns out to be easy to extend to higher dimensions.

First, we select the positive integers n and m and divide the intervals $(a, b) \times (c, d)$ into $(n + 1) \times (m + 1)$ grid points including points on the boundary with grid point (i, j) located at $(x_i, y_j) = (a + ih, b + jk)$ for $0 \leq i \leq n$ and $0 \leq j \leq m$. In this way we impose a cartesian grid (or mesh) with grid spacing $h = (b - a)/n$ in the x-direction and $k = (d - c)/m$ in the y-direction. The boundary points Γ are defined as the set of mesh points in $\mathbb{R}^2$ which don't belong to Ω, but which have a nearest neighbour in Ω. In the so-called vertex-centered discretisation grid points are placed at the vertices. In the so-called cell-centered discretisation, the grid points are placed at the center of the grid cells so that there are $n \times m$ grid points (none lying on the boundary) and the grid point (i, j) is located at $(x_i, y_j) = (a + \frac{2i-1}{2}h, c + \frac{2j-1}{2}k)$ for $1 \leq i \leq n$ and $1 \leq j \leq m$. The interior of the discrete grid is denoted by Ω_h and the boundary by Γ_h or $\partial\Omega_h$. Fig. 2.6 shows examples of vertex and cell-centered discretisations of a square domain.



(a) Vertex-centred (b) Cell-centred

Figure 2.6: Vertex-centred and cell-centred discretization of a square domain.

Once the grid is in place the operators in the PDE can be approximated locally using Taylor's series expansion. Using this expansion we can approximate the operator $\frac{\partial u}{\partial x}$ at the grid point (i, j) in 3 ways, the first order forward and backward difference operators defined respectively by

$$\frac{\Delta_x^+(u)_{i,j}}{h} \approx \frac{u(x+h,y)-u(x,y)}{h} = \frac{(u)_{i+1,j}-(u)_{i,j}}{h}$$

and

$$\frac{\Delta_x^-(u)_{i,j}}{h} \approx \frac{u(x,y)-u(x-h,y)}{h} = \frac{(u)_{i,j}-(u)_{i-1,j}}{h}$$

or the second order central difference approximation

$$\frac{\Delta_x^c(u)_{i,j}}{2h} \approx \frac{u(x+h,y)-u(x-h,y)}{2h} = \frac{(u)_{i+1,j}-(u)_{i-1,j}}{2h},$$

where $(u)_{i,j} = u(x_i, y_j)$ is the value of $u(x, y)$ at the grid point (i, j). Similarly using the Taylor expansion, the approximations to higher order derivatives can be constructed in a similar way. A second order approximation to $\frac{\partial^2 u}{\partial x^2}$ at (i, j) is given by

$$u''_{xx}(x,y) \approx \frac{u(x+h,y)-2u(x,y)+u(x-h,y)}{h^2}$$

$\frac{\partial^2 u}{\partial y^2}$ can be defined in a similar way.

In this way, substituting the finite difference approximation to the continuous problem on the discrete domain, which is denoted by

$$L_h^\Omega u_h(\mathbf{x}) = f_h^\Omega(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Omega_h \tag{2.39}$$

$$L_h^\Gamma u_h(\mathbf{x}) = f_h^\Gamma(\mathbf{x}) \text{ for } (\mathbf{x}) \in \Gamma_h \tag{2.40}$$

gives an approximation to the given problem with a truncation error equal to the order of the finite difference approximation. In the above notation u_h is a grid function on $\Omega_h \cup \Gamma_h$, L_h^Ω and L_h^Γ are operators on the space of grid functions and f_h^Ω and f_h^Γ

are discrete representations of f^{Ω} and f^{Γ}. Usually the boundary conditions can be eliminated and (2.39) and (2.40) can be written simply as

$$L_h u_h = f_h. \tag{2.41}$$

Example 2.5.3 *Let us consider the Poisson equation (2.33). For simplicity we will consider a very simple domain on the unit square with Dirichlet boundary conditions (2.34). Assume that the domain is discretised using a vertex-centered grid with $h = k = 1/n$ then at interior grid points which are not adjacent to the boundary a second order central difference approximation is given by*

$$(L_h u_h)_{i,j} = \frac{-u_{i+1,j} - u_{i-1,j} + 4u_{i,j} - u_{i,j+1} - u_{i,j-1}}{h^2} = (f_h^{\Omega})_{i,j}. \tag{2.42}$$

The scheme is a 5-point difference operator scheme. At points adjacent to the right boundary, for example, $(u)_{n+1,j}$ will be replaced by the boundary value $(f_h^{\Gamma})n, j$, i.e.

$$(L_h u_h)_{n,j} = \frac{-u_{n-1,j} + 4u_{n,j} - u_{n,j+1} - u_{n,j-1}}{h^2} = (f_h^{\Omega})_{n,j} + \frac{(f_h^{\Gamma})_{n,j}}{h^2}. \tag{2.43}$$

Since $(f_h^{\Gamma})_{n,j} = 0$ we have

$$(L_h u_h)_{n,j} = \frac{-u_{n-1,j} + 4u_{n,j} - u_{n,j+1} - u_{n,j-1}}{h^2} = (f_h^{\Omega})_{n,j}. \tag{2.44}$$

Similar considerations give $L_h u_h$ at other points adjacent to the boundary, therefore we have $L_h u_h = f_h$ where u_h is a grid function on the interior grid points only.

We discetise similarly for problems with Neumann boundary conditions.

2.5.1 Boundary Conditions

Boundary conditions define the set of conditions specified for behavior of the solution to a set of differential equations at the boundary of its domain. So far we have only mentioned Dirichlet boundary conditions on vertex-centered grids, out of 13 different categories such as Neumann boundary conditions, Cauchy boundary conditions, Mixed boundary conditions, Periodic boundary conditions. We briefly now describe how to deal with Neumann boundary conditions with vertex and cell-centered grid since this will be used for the rest of the thesis.

Neumann Boundary Conditions for Vertex-centered Grids

The Neumann boundary condition is a type of boundary imposed on an ordinary or a partial differential equation, specifying the values that the derivative of a solution is to take on the boundary of the domain. Let us assume that we have a Neumann boundary condition $\frac{\partial u}{\partial n}(x, y) = f^{\Gamma}(x, y)$ on the right boundary of a vertex centered grid. We assume that the discrete equation $L_h u_h(x, y) = f_h(x, y)$ extends to the points on the right boundary. The equation at these grid points will involve ghost grid points

outside the domain. These ghost grid points can be eliminated using the Neumann boundary condition

$$\frac{(u)_{n+1,j} - (u)_{n-1,j}}{2h} = (f^{\Gamma})_{n,j}.$$

Example 2.5.4 *Going back again to the example of Poisson's equation (2.33) the unit square then at the right boundary, we have*

$$(L_h u_h)_{n,j} = \frac{-2u_{n-1,j} + 4u_{n,j} - u_{n,j+1} - u_{n,j-1}}{h^2} = (f_h^{\Omega})_{n,j} + \frac{2}{h}(f_h^{\Gamma})_{n,j}. \tag{2.45}$$

Neumann Boundary Conditions for Cell-centered Grids

In the case of a cell-centered grids we have no points on the boundary, so in general the equation at interior points which are adjacent to the boundary will involve ghost points outside of the domain, which need to be eliminated using the boundary condition. If we have a Neumann boundary condition at this right boundary, for example, we can write it as

$$\frac{(u)_{n+1,j} - (u)_{n,j}}{h} = (f^{\Gamma})_{n+1/2,j}.$$

2.5.2 Nonlinear Equations

Nonlinear PDEs are more difficult to study and there are almost no general techniques that work for all such equations, but they are more attractive since they describe many different phenomena. Nonlinear PDEs have been a greatly successful in image processing for solving problems such as denoising, segmentation, etc.

Nonlinear PDEs are treated in much the same way as linear equations. The various operators in the equation are approximated locally on a discrete grid using the finite difference method. The discrete nonlinear equation is denoted by

$$N^{\Omega}u(\mathbf{x}) = f^{\Omega}(\mathbf{x}) \text{ for } (\mathbf{x},) \in \Omega \tag{2.46}$$

$$N^{\Gamma}u(\mathbf{x}) = f^{\Gamma}(\mathbf{x}) \text{ for } (\mathbf{x}.) \in \Gamma \tag{2.47}$$

Similarly, the boundary conditions are usually eliminated and then the discrete nonlinear equation can be written simply as

$$N_h(u_h) = f_h. \tag{2.48}$$

It may be possible to write the nonlinear equation in matrix notation, e.g. $A_h(u_h)u_h = f_h$, where the some of the matrix entries will depend on u_h. Examples of using finite difference methods for specific nonlinear PDEs will be shown in subsection §2.6.1 and later in Chapter 3 followed by all the other chapters.

2.6 Level Set Methods

A level set method is a numerical technique which helps with tracking moving fronts to interfaces and shapes. This technique was first introduced by Osher and Sethian in [122], where the boundaries are given by level sets of a function $\phi(\mathbf{x})$, naming it as the Level Set Method. This method is very successful due to a very easy way of following shapes that change topology, for example when a shape splits or merges in two. The great success of level set methods can be attributed to the role of curvature, rising according to the curve propagation speed function, approaching numerical solutions of a time-dependent equation for a moving implicit surface such that the proper solution is obtained. More about the connection between curvature and the notion of entropy conditions was explored by Sethian [147]. The application of level set methods to a wide variety of problems including image processing, fluid dynamics flows, visualization, computer vision, control, visibility, segmentation, restoration and many others cen be found in [108, 149, 151, 150].

In this section, we introduce the level set method applied to the theory of curve/ surface evolution and we will show the efficiency of solving the problem of moving fronts using the level set formulation of a curve/surface, in particular the problem of changes of topology, and in the following sections we will discuss this similar work along with many of the auxiliary equations that were developed in recent years.

For a given interface $\Gamma = \partial\Omega$ the level set is independent of the parametrisation of the contour and can be used to represent the interface evolution. The idea of the level set method is to implicitly represent an interface Γ in $\mathbb{R}^2$ (same Γ in $\mathbb{R}^n$) as the level set of a function ϕ, called the level set function of higher dimension (in this case $\mathbb{R}^3$) and compute the geometric characteristics and the motion of the front with this level set function. The level set function ϕ of the closed front Γ is defined as follows [121]:

$$\begin{cases} \phi(\mathbf{x}) > 0 \text{ inside } \Gamma \\ \phi(\mathbf{x}) < 0 \text{ outside } \Gamma \\ \phi(\mathbf{x}) = 0 \text{ on } \Gamma \end{cases},$$

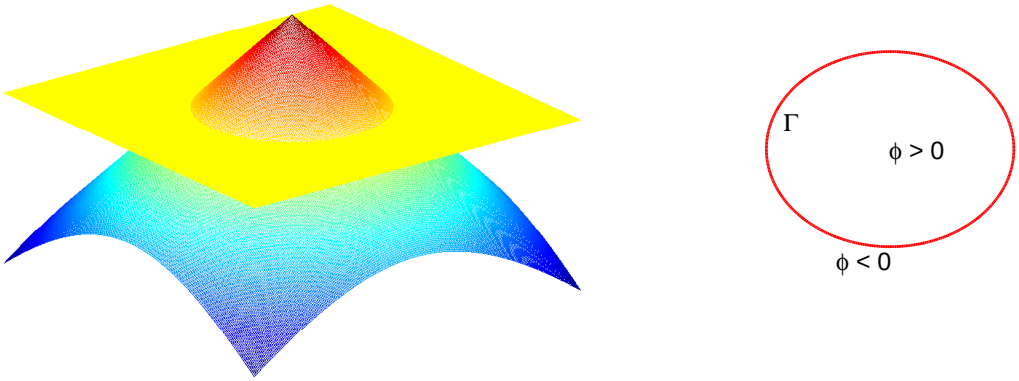
$\mathbf{x} \in \mathbb{R}^2$.

Adjusting the contour at time t denoted by $\phi(\mathbf{x}(t); t)$

$$\begin{cases} \phi(\mathbf{x}(t);t) > 0 \text{ inside } \Gamma \\ \phi(\mathbf{x}(t);t) < 0 \text{ outside } \Gamma \\ \phi(\mathbf{x}(t);t) = 0 \text{ on } \Gamma. \end{cases}$$

A level set model, for a given time t specifies a surface in an implicit form as an iso-surface of a scalar function $\phi : \mathbb{R}^3 \to \mathbb{R}$, embedded in $\mathbb{R}^3$, i.e.,

$$S := \{\mathbf{x} : \phi(\mathbf{x}) = k\},$$



(a) The level set function ϕ (b) The interface Γ

Figure 2.7: Plot of the higher dimension (3-D) function $\phi(x)$ and the interface $\Gamma = \{(x, y) \in \Omega : \phi(x, y) = 0\}$.

where $\mathbf{x}$ is a point in space on the iso-surface and k is the iso-value and is arbitrary.

Assume that the normal velocity $v(\mathbf{x})$ is known for every point $\mathbf{x}$ with $\phi(\mathbf{x}) = 0$ on the implicit given interface Γ. The simplest way to move all the points on the surface with this velocity is to solve the ordinary differential equation (ODE)

$$\begin{cases} \frac{d\mathbf{x}(t)}{dt} = v(\mathbf{x}) \\ \Gamma(t = 0) = \Gamma_0 \end{cases} \tag{2.49}$$

for every point $\mathbf{x}$ on the front $\Gamma = \partial\Omega$, i.e., for all $\mathbf{x}$ with $\phi(\mathbf{x}) = 0$. This is the Lagrangian formulation of the interface evolution equation.

In order to avoid problems with the deformation of surface elements and complicated topology of interfaces, the implicit function of the level set $\phi(\mathbf{x})$ can be used. The level set value of a point on the contour with motion $\mathbf{x}'(t)$ must always be 0 [3],

$$\phi(\mathbf{x}(t); t) = 0.$$

By the chain rule we obtain

$$\phi_t + \nabla\phi(\mathbf{x}(t), t) \cdot \mathbf{x}_t = 0, \tag{2.50}$$

which is a Hamilton-Jacobi equation. Denoting by F the speed in the outward normal direction, $\mathbf{n} = \nabla\phi/|\nabla\phi|$ the outward normal vector and considering the equality $\nabla\phi \cdot \nabla\phi = |\nabla\phi|^2$ we obtain

$$F = \mathbf{x}_t \cdot \mathbf{n} = \mathbf{x}_t \cdot \frac{\nabla\phi}{|\nabla\phi|} \Leftrightarrow F|\nabla\phi| = \mathbf{x}_t \cdot \nabla\phi \tag{2.51}$$

[3] Actually propagation is considered for all the level sets, not just $\phi(x) = 0$.

Substituting 2.51 into 2.50 yields

$$\phi_t + F\ |\nabla\phi| = 0$$

and the level set equation is given by the evolution of the zero level set of $\phi(\mathbf{x})$ which is a solution of the following PDE:

$$\begin{cases} \frac{\partial\phi(\mathbf{x},t)}{\partial t} = -F(\mathbf{x},t)|\nabla\phi(\mathbf{x},t)| \\ \phi(\mathbf{x},t=0) = d(\Gamma_0) = \phi_0(\mathbf{x}), \end{cases} \tag{2.52}$$

where d is a function (usually a signed distance function, see §2.6.3) whose zero level set is the initial contour Γ_0. In general $F(\mathbf{x})$ is defined on the entire domain Ω, but for numerical purposes it is sufficient to be defined only on a band containing the interface.

In a wide range of important applications the speed function is considered as [121] $F(\mathbf{x}) = F_0(\mathbf{x}) - \varepsilon(\mathbf{x})\kappa(\mathbf{x})$. The first term $F_0(\mathbf{x})$ is a convection term. In the active contour framework, for example in geodesic active contour [28], the function $F_0(\mathbf{x})$ represents either an attraction force toward the boundaries of objects, i.e $F_0(\mathbf{x}) = \nabla g$ where g is an edge detector function usually given by $g(\nabla(z*G_\sigma)) = \dfrac{1}{1+\gamma|\nabla(z*G_\sigma)|^2}$, or a balloon force (further details are given in the next chapter). The second term $\varepsilon(\mathbf{x})\kappa(\mathbf{x})$ is a contour smoothing term based on the curvature $\kappa(\mathbf{x})$ of level sets of $\phi(\mathbf{x})$ and γ a scalar. The above equation can be rewritten as follows

$$\phi_t(\mathbf{x}) = -F_0(\mathbf{x})|\nabla\phi(\mathbf{x})| + \varepsilon(\mathbf{x})\kappa(\mathbf{x})|\nabla\phi(\mathbf{x})| \tag{2.53}$$

From equation (2.49) and the above discussion we can notice that:

- The equations of motion are independent of the choice of the initial level set function ϕ_0 [43, 54, 55, 56]. This reflects the fact that only level sets matter, not the point values of the representing function;
- The zero level set of Γ and all its level sets follow the front evolution equation (2.52);
- The classical solution, if it exists, of (2.52) coincides with the classical solution (2.49);
- Level set is a parametrization free formulation;
- Singularities can arise with PDEs (2.49).

The level set method allows topological changes, such as the breaking or merging of curves as illustrated in Fig. 2.8. In the upper row of the figure we see a shape in dark red; that is, a bounded region with a well-behaved boundary, in red. Below it, the blue surface is the graph of a level set function ϕ determining this shape, and the flat yellow region represents the x-y plane. The boundary of the shape is then the zero level set of ϕ, while the shape itself is the set of points in the plane for which ϕ is positive or zero.

We can clearly see the shape changing its topology by splitting in two. Describe this transformation numerically by parameterising the boundary of the shape and following its evolution it would be hard. One would need an algorithm able to detect the moment of the shape splitting in two, and then construct parameterisations for the two newly obtained curves. On the other hand, we see that the level set function can easily work with a shape by merely getting translated.

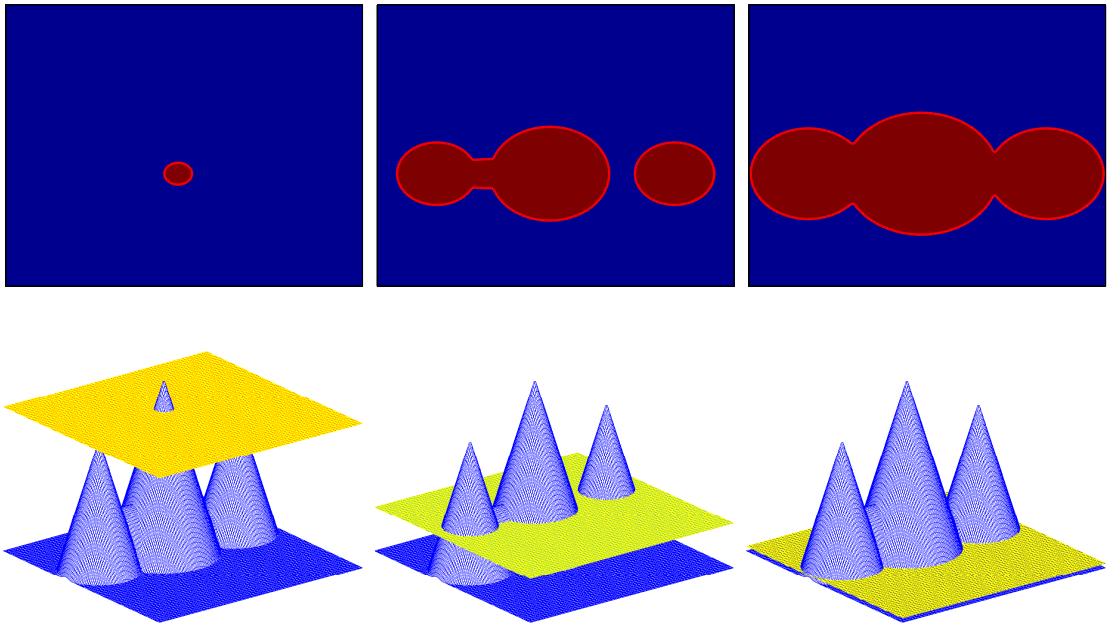


Figure 2.8: Topological changes of the level set function. The top row presents the evolution of the curve and the bottom row shows the evolution of the associated level set function. We see that the curve changed its topology but not the level set function.

2.6.1 Numerical Implementation of the Level Set Method

In this section we discuss the numerical implementation of a Hamilton-Jacobi equation (2.52) with $F(\mathbf{x})$ given in (2.53). We consider the solution in a two dimensional space, so that $\mathbf{x} = (x, y) \in \mathbb{R}^2$, and equation (2.53) is written:

$$\begin{cases} \frac{\partial \phi(x,y)}{\partial t} = -F_0(x,y)|\nabla\phi(x,y)| + \varepsilon(x,y)|\nabla\phi(x,y)|\nabla \cdot \left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right) \\ \phi(x,y;t=0) = d(\Gamma_0) = \phi_0(x,y). \end{cases} . \tag{2.54}$$

Using the first-order Taylor expansion with respect to t, for a small time interval Δt we have

$$\phi(x,y,t+\Delta t) = \phi(x,y;t) + \Delta t \phi_t(x,y,t) \tag{2.55}$$

and in this way the value of ϕ_t is approximated. To approximate the derivatives of the first term on the right hand side of equation (2.54), $F_0|\nabla\phi|$, we can use an entropy

satisfying scheme and approximate the derivative of the second term using a central difference approximation.

The forward, backward and central differences, $\triangle^{\pm}$ and $\triangle^{0}$ as a shorthand, representing the partial derivative of ϕ with respect to both directions x and y, are defined by the familiar expressions

$$\triangle_x^{\pm}\phi = \pm\frac{\phi(x \pm h_1, y) - \phi(x, y)}{h_1}, \triangle_y^{\pm}\phi = \pm\frac{\phi(x, y \pm h_2) - \phi(x, y)}{h_2},$$

$$\triangle_x^{0}\phi = \frac{\phi(x + h_1, y) - \phi(x - h_1, y)}{2h_1}, \triangle_y^{0}\phi = \frac{\phi(x, y + h_2) - \phi(x - h_2, y)}{2h_2},$$

$$\triangle_{xx}^{0}\phi = \frac{\phi(x + h_1, y) - 2\phi(x, y) + \phi(x - h_1, y)}{h_1^2}, \triangle_{yy}^{0}\phi = \frac{\phi(x, y + h_2) - 2\phi(x, y) + \phi(x - h_2, y)}{h_2^2},$$

and

$$\triangle_{xy}^{0}\phi = \frac{\phi(x + h_1, y + h_2) + \phi(x - h_1, y - h_2) - \phi(x + h_1, y - h_2) - \phi(x - h_1, y + h_2)}{4h_1h_2},$$

were h_1 and h_2 are horizontal and vertical spatial step sizes respectively.

Entopy Satisfying Scheme. If our speed function were such that $F(0) = 0$, or had no constant part, then the entropy satisfying scheme would not be necessary. In the absence of the curvature term the solution can develop shocks, which are discontinuities defined by jumps in derivatives. Special numerical methods are necessary to handle these discontinuities. A highly robust and accurate computational method to solve Eq.(2.54) was developed by Osher and Sethian [122] based on the notion of weak solutions and entropy limits, a so-called "up-wind scheme", with the essentially non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) schemes in [123, 81]. When a spatial point may be a shock, it becomes important to look separately at its forward and backward derivatives. If the forward derivative has a different sign to the backward derivative, then the curve is either flowing out or into our point, or if a curve intersecting that point contains a shock at that point and if the speed function is positive instead of negative. To handle these we can discretise using the following numerical scheme:

$$F_0(\mathbf{x})|\nabla\phi| \approx \max(F_{0ij}, 0)\nabla^{+} + \min(F_{0ij}, 0)\nabla^{-}, \tag{2.56}$$

with

$$\nabla^{+} = \sqrt{((\triangle_x^{-}\phi)^{+})^2 + ((\triangle_x^{+}\phi)^{-})^2 + ((\triangle_y^{-}\phi)^{+})^2 + ((\triangle_y^{+}\phi)^{-})^2}$$
$$\nabla^{-} = \sqrt{((\triangle_x^{+}\phi)^{+})^2 + ((\triangle_x^{-}\phi)^{-})^2 + ((\triangle_y^{+}\phi)^{+})^2 + ((\triangle_y^{-}\phi)^{-})^2},$$

and $(\cdot)^{+} = \max(\cdot, 0), (\cdot)^{-} = \min(\cdot, 0)$.

Figure 2.9 illustrates this numerical scheme for $F_0(x, y) = \pm 1$ for a circular curve

propagating inward and outward. The blue curve is the original curve and the green curve is the propagated curve after applying the numerical scheme (2.56) on the balloon term.

Figure 2.9: Evolution of a circular contour inward (left) and outward (right). The left image shows propagating inward, the blue contour is the initial and the green contour is the final propagated contour, given $F(x, y) = -1$. The right image shows propagating outward, the blue contour is the initial and the green is the final propagated contour, given $F(x, y) = 1$. These results are obtained by implementation of equation (2.56).

Central Difference Approximation Scheme. The second term in the right hand side of equation (2.54) is a regularization term based on the mean curvature. This term is parabolic and therefore does not need an upwind scheme, which is designed for hyperbolic advection terms to guide the propagation directions. For a parabolic term, the propagation is in all directions, hence the central difference approximation scheme is suitable to approximate the term $\varepsilon|\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ with first order accuracy. In 2-D images, the term $|\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ of the level sets is equal to

$$|\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)=\frac{\phi_{xx}\phi_y^2-2\phi_x\phi_y\phi_{xy}+\phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{1/2}} \tag{2.57}$$

and the associated numerical scheme for the second term is

$$\varepsilon|\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)=\varepsilon\frac{\triangle_{xx}^0\phi(\triangle_y^0\phi)^2-2\triangle_x^0\phi\triangle_y^0\phi\triangle_{xy}^0\phi+\triangle_{yy}^0\phi(\triangle_x^0\phi)^2}{((\triangle_x^0)^2+(\triangle_y^0)^2)^{1/2}}, \tag{2.58}$$

Figure 2.10 shows the effect of the smoothing term tested on a non smooth curve, where the left part is the initial contour of the level set function ϕ_0 and the right part is the smooth contour, when regularization term is applied.

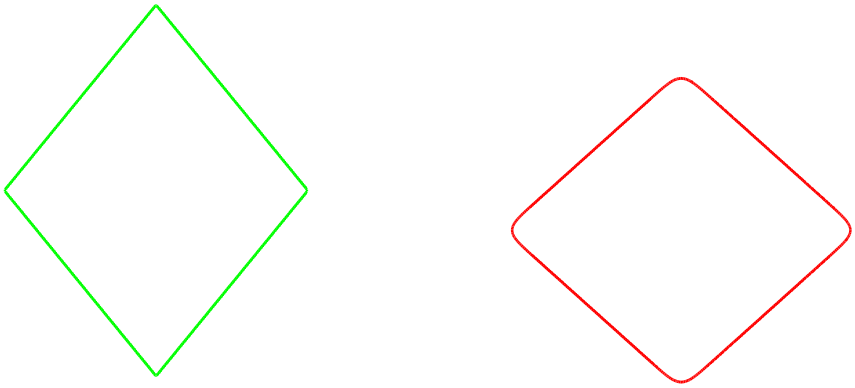


Figure 2.10: Smoothing of contours represented by the zero level set of a level set function. The left image displays the initial contour ϕ_0, while the right image the smooth contour after applying the curvature dependent regularizer.

2.6.2 Distance Function

In mathematics, a metric or distance function is a function which defines a distance between elements of a set. There are several distance functions which meet these requirements and have been used. The most commonly used distance is the Euclidian distance defined as

$$d(\mathbf{w}) = \min(|\mathbf{w} - \mathbf{w_I}|) \text{ for all } \mathbf{w_I} \in \mathcal{A}, \tag{2.59}$$

where $\mathcal{A}$ is a set of given points. In case of a given curve, considering the boundary $\mathcal{A} = \partial\Omega$, the distance implies that $d(\mathbf{w}) = 0$ on the boundary where $\mathbf{w} \in \partial\Omega$. Geometrically, d may be constructed as follows. If $\mathbf{w} \in \partial\Omega$, then $d(\mathbf{w}) = 0$. Note that, since d is Euclidean distance,

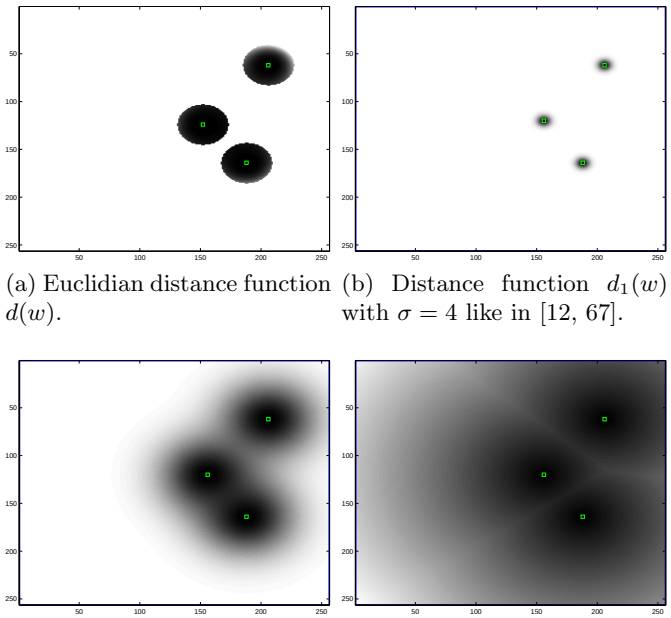
$$|\nabla d| = 1.$$

There are several distance functions which meet the above requirement and have been used as constraint functions in imaging (see [180, 12, 67]). For a given finite domain $V = \Omega$ and a set of points $\mathcal{A} = w_i^* : i = 1, ..., n_1 \in \Omega$ the distance function d can be defined in the following way [67, 12] :

$$d_1(w) = \prod_{i=1}^{n_1} \Big(1 - \exp(-\frac{|w - w_i^*|^2}{2\sigma^2})\Big). \tag{2.60}$$

where σ is a positive number. In Zhang-Chan [180] another option for d has been proposed. They change the parameter σ into $\bar{\sigma}$, i.e.

$$\bar{\sigma} = \min_{1 \le i < j \le n_1} |w_i^* - w_j^*|, \tag{2.61}$$



(a) Euclidian distance function $d(w)$.

(b) Distance function $d_1(w)$ with $\sigma = 4$ like in [12, 67].

(c) Distance function $d_2(w)$ as proposed in [180].

(d) Non-smooth distance function $d_3(w)$.

Figure 2.11: Comparison of several distance functions.

and define a new function as follows:

$$d_2(w) = \prod_{i=1}^{n_1} \Big(1 - \exp(-\frac{|w - w_i^*|^2}{2\bar{\sigma}^2})\Big). \tag{2.62}$$

Gout et al. in [67] proposed the distance function

$$d_3(w) = \frac{\min\limits_{w_i^* \in \mathcal{A}} |w - w_i^*|}{\max\limits_{w_i^* \in \mathcal{A}} \max\limits_{\bar{x} \in \Omega} |\bar{x} - w_i^*|} = \frac{1}{M} \min_{w_i^* \in \mathcal{A}} |w - w_i^*|. \tag{2.63}$$

Clearly d acts locally and will be approximately 0 in the neighborhood of points of $\mathcal{A}$. In Fig. 2.11, we test the above distance function $d_1(w)$, $d_2(w)$, $d_3(w)$ and $d_4(w)$. With domain size $\Omega = 256 \times 256$, the black region indicates that the distance function approaches to 0, the white region is close to 1 and green circles present the location of the given points.

2.6.3 Signed Distance Functions

In subsection §2.2 we defined implicit functions with $\phi(\mathbf{x}) \leq 0$ in the interior region Ω^-, $\phi(\mathbf{x}) > 0$ in the exterior region Ω^+, and $\phi(\mathbf{x}) = 0$ on the boundary $\partial\Omega$. In this subsection we discuss signed distance functions, which are a subset of the implicit functions defined in Section §2.2.

For a given curve (surface) in real space subdomain Ω and its interface $\partial\Omega$, a signed distance function is an implicit function ϕ with the property $|\phi(\mathbf{x})| = d(\mathbf{x})$ for all $\mathbf{x}$, in

other words $\phi(\mathbf{x}) = \pm\, d$; for all $\mathbf{x} \in \mathbb{R}^n$.

Thus, our ϕ takes on the distance from the boundary with a sign depending on being inside or outside the region,

$$\phi(\mathbf{x}) = \begin{cases} -d(\mathbf{x}) & \mathbf{x} \in \Omega^- \\ 0 & \mathbf{x} \in \partial\Omega \\ d(\mathbf{x}) & \mathbf{x} \in \Omega^+ \end{cases} .$$

There are a number of properties of this specific definition of ϕ which help with the numerics of solving this level set problem. Any function which is negative inside and positive outside the initial front would work, but the signed distance function gives many nice properties. The main property is:

$$|\nabla\phi(\mathbf{x})| = 1. \tag{2.64}$$

This is true for all points except if they are equidistant from at least 2 points on the interface. This set of points is a zero measure set. The unit normal vector for a sign distance function is $\mathbf{n} = \nabla\phi$ and curvature $\kappa = \Delta\phi$.

Example 2.6.1 *In two dimensional space the circle with center point a, b and radius r with equation*

$$\phi(\mathbf{x}) = \sqrt{(x-a)^2 + (y-b)^2} - r$$

or in three dimensional space the sphere with center point a, b, c and radius r

$$\phi(\mathbf{x}) = \sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2} - r.$$

2.6.4 Re-initialization

Large variations in $\nabla\phi$ for general speed functions F, might increase the error when calculating the derivatives numerically or as the interface evolves, ϕ will generally drift away from its initialized level set [155]. Further small changes in ϕ might lead to a large change of the zero level set if the level set function is too flat around zero. Mulder, Osher, and Sethian [114] demonstrated that initializing ϕ to a signed distance function $\phi = \pm d$ results in more accurate numerical solutions. In order to reduce the numerical errors caused by both steepening and flattening effects, Chopp [44] introduced the idea of reinitializing the level set function periodically throughout the calculation. Since only the $\phi = 0$ isocontour has any meaning, one can stop the calculation at any point in time and reset the other isocontours so that ϕ is again initialized to a signed distance function which means to change ϕ into the distance function such that $|\nabla\phi| = 1$ for all $\mathbf{x}$.

Re-initialization does not change the mathematical problem, since re-initialization changes the level set function everywhere, except from the zero level set which must be the same before and after re-initialization and in the same way since the regions of interest are where $\phi > 0$ and where $\phi < 0$, these areas do not change through the

re-initialization.

Based on the Rouy and Tourin numerical method [134] for solving $|\nabla\phi| = f(\mathbf{x})$, reinitialization algorithms maintain the signed distance property by solving to steady state (as time $t \to \infty$) of the following PDE for ϕ

$$\begin{aligned} &\phi_t - \text{sgn}(\phi)(1 - |\nabla\phi|) = 0 \qquad \text{in } \mathbb{R} \times (0, \infty), \\ &\phi(\mathbf{x}, 0) = \phi_0 \qquad \text{in } \mathbb{R}, \end{aligned} \tag{2.65}$$

where ϕ_0 is the function which is supposed to be re-initialized and $\text{sgn}(\cdot)$ is a one-dimensional signed distance approximated numerically in [155]

$$sgn(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}}.$$

Later, Peng, Merriman, Osher, Zhao, and Kang [125], suggested a better choice for signum function, especially when the initial ϕ_0 was a poor estimate of signed distance.

2.7 Regularization of Ill-Posed Problem

The inverse and ill-posed problems have been steadily and surely gaining popularity in the last fifty years and important results are still being discussed and attempts being made to improve them. Problems of this kind have been shown in various branches of classical mathematics (computational algebra, partial differential equations, differential and integral equations, functional analysis) and they are among the most complicated (due to their instability and, usually, nonlinearity). They have been studied and applied systematically in physics, geophysics, astrophysics, image processing and other areas of knowledge where mathematical methods are used. Usually regularization of the inverse ill-posed problem is required to make it well-posed and therefore solvable. In this section, we review the very basic theory of inverse and ill-posed problems and the concept of regularization introduced by A. N. Tikhonov in 1963 ([157, 158, 159]). Regularization is based on the idea of approximating an ill-posed problem by a number of well-posed problems, which are "close" to the original one.

2.7.1 Inverse Problems and Ill-posedness

An inverse problem is a general task that is used to convert observed measurements into input data or parameter values of the physical model. If in forward (direct) modeling problems we try to find exact or approximate functions that describe various phenomena (with some known properties), for inverse problems results of experiments (forward modeling) are considered as data and the unknown input as the solution of the model. The inverse problem is the inverse of the forward problem.

Definition 2.7.1 (**Well-posed Problem**). *According to Hadamard [70], a problem is well-posed (or correctly-set) if:*

1. *the solution exists (solvability condition),*
2. *the solution is unique (uniqueness condition),*
3. *the solution depends continuously on the data and parameters (stability condition).*

The meaning of (2.) is "unique within a certain class of functions". For example, a problem which has several solutions, only one of which is bounded is said to have a unique solution in the space of bounded functions. And (3) means that "small" changes in initial or boundary functions and in parameter values result in "small" changes in the solution (in some appropriate norm). If the problem is well-posed, then it stands a good chance of using a stable algorithm for a computer solution.

Definition 2.7.2 (**Ill-posed Problem**). *Problems that are not well-posed in the sense of Hadamard are termed ill-posed [70, 159].*

Often, the ill-posedness of certain practical problems is due to their lack of precise mathematical formulation. Ill-posed problems are usually understood as those results where small changes in the assumptions lead to arbitrarily large changes in the conclusions.

Inverse problems are typically ill-possed, with the stability condition of well-posedness most often violated.

Example 2.7.3 *The classical example to illustrate an inverse problem is the Fredholm integral equation:*

$$g(s) = \int_{\alpha_1}^{\alpha_2} K(s,t)u(t)\,dt,\ \beta_1 \leq s \leq \beta_2 \tag{2.66}$$

where $g(s)$ is a known bounded function, the left-hand side function $K(s,t)$ is a smooth bounded function (known or unknown function of two variables s and t), while $u(t)$ is the unknown function that we wish to compute.

If the solution $u(t)$ is perturbated by $\Delta u(t) = \eta \sin(2\pi pt)$, $p = 1, 2, ...,$ with η a constant then the perturbation on the left-hand side g is given by

$$\Delta g(s) = \eta \int_{\alpha_1}^{\alpha_2} K(s,t)\sin(2\pi pt)\,dt,\ p = 1, 2, ...,$$

and due to Riemann-Lebesgue lemma it follows that $\Delta g(s) \to 0$ as $p \to \infty$ [26]. Hence small errors in the measured data $g(s)$ will introduce an arbitrary large ratio $||\Delta u||/||\Delta g||$, by choosing p large enough. In this sense the inverse problem 2.66 is ill-posed.

Example 2.7.4 *Approximating the solution u for given y and a bounded operator $A : U \to Y$ (U and Y Hilbert spaces) such that*

$$Au = y,$$

represents an ill-posed problem.

Example 2.7.5 *In image reconstruction and signal processing, reconstruction of the image u from possibly noisy data $y \in R^N$ is required, supposing that the relation between them is given by Fredholm integral equation:*

$$y_i = \int_D K(s_i, t)u(t)\ dt + \eta_i,$$

for the measured image data $y = (y_1, ..., y_N)^d \in R^N$ with noise η_i assumed to be Gaussian with mean 0 and standard deviation σ_i, with $K(s_i, t)$ a smooth kernel (see subsection 2.7.3), and D is the domain of integration. By the quadrature rule we can rewrite as

$$\int_D K(s_i, t)f(t)\ dt = \sum_{j=1}^{M} w_j K(s_i, t_j)u(t_j)\Delta t_j,$$

and the above equation can be written in discrete form

$$y = Au + \eta,$$

where $A_{ij} = w_j K(s_i, t_j)\ t_i$ and $u = u(t_j)$ (for $A = I$ the image reconstruction problem is called denoising otherwise deblurring). We can see that the problem is ill-conditioned since the data contains noise. Therefore a regularization method is needed for the solution of the problem.

2.7.2 Regularization

Regularization is a technique used to transform an inverse problem into a well-posed problem. Tikhonov *et al.* [157, 159] introduced a popular way to overcome ill-posed minimization problems. The basic idea is to introduce a new constraint to the problem which demands the solution to belong to a specific set of solutions or to have specific features, represented within a regularizing functional. Regularization involves additional information in order to solve an ill-posed problem and prevent over-fitting. This information is usually of the form of a penalty for complexity, such as restrictions for smoothness or bounds on the vector space norm.

Let us consider a given $K : D(K) \subseteq U \to Y$ operator between Hilbert spaces U and Y, such that $Ku = y$, when $u^* = K^{-1}y$ does not exist or even if it exists[4], it might not be computable or non-unique. Usually instead of exact data $y = Ku$ we have measured noisy data $y_\delta = Ku + \delta$, such that $\|y - y_\delta\| \leq \delta$.

A standard treatment for this ill-posedness is to look at a generalized solution defined by a solution to the least square problem

$$u^* = argmin_{u \in \mathcal{D}(F)} \left(\|Ku - y_\delta\|^2 \right). \tag{2.67}$$

Important advantages can be obtained by relating specific required properties of the solution [53, 87]. All such methods fall naturally within the framework of the regularizing functional. To obtain a feasible numerical algorithm it has to be approximated by

[4] u^* represents a generalized solution, an approximation of real solution u

a sequence of stable operators, the regularization operators approximate an ill-posed problem with a family of well-posed problems. In 1963, Tikhonov introduced a stable method for numerically computing solutions to inverse problems. Approximate the inverse K^{-1} by a family of stable operators R_α such that the problem of $u^* = K^{-1}(y)$ is almost equal to $u^*_\alpha = R_\alpha(y) \Rightarrow K \sim R_\alpha$, where R_α are called regularization operators and $\lim_{\alpha\to 0} R_\alpha(y) = K^{-1}(y)$ with α a regularization parameter. He proposed to minimize the Tikhonov functional

$$u^*_\alpha = argmin_{u\in\mathcal{D}(K)}\left(\|Ku - y_\delta\|_2^2 + \alpha^2\|u\|_2^2\right). \tag{2.68}$$

This technique can be generalized by choosing other norms or seminorms as penalty functional and rewrite the generalized Tikhonov regularization [53]:

$$u^*_\alpha = argmin_{u\in\mathcal{D}(K)}\left(\|Ku - y_\delta\|_{L^2}^2 + \alpha^2 J(u)\right). \tag{2.69}$$

One important choice in image processing is the bounded variation seminorm, which leads to the Rudin-Osher-Fatemi method (ROF) in their seminal 1992 paper [135] introduced the Total Variation (TV) regularization functional

$$|u|_{TV} = \int_\Omega |\nabla u| dxdy, \tag{2.70}$$

where Ω is the image domain. Tikhonov regularization with the TV regularization functional can remove noise while still preserving the edges in an image. Note that the Euclidean norm $|\cdot|$ is not differentiable at the origin. To overcome the resulting numerical difficulties, (2.70) is replaced by the following

$$|u|_{TV_\beta} = \int_\Omega \sqrt{|\nabla u|^2 + \beta^2} dxdy, \tag{2.71}$$

where $\beta > 0$ is a small perturbing parameter. More about TV regularization can be found in [154] and in a wide range of applications in image processing problems, see [75, 77, 76, 135].

2.7.3 Convolution

Convolution is both a *mathematical concept* and an *important tool in data processing*, in particular in digital signal and image processing. From a mathematical viewpoint, convolution is a mathematical operation on two functions f and g, producing a third function that is typically viewed as the amount of overlap of the function g as it is shifted over the other function, f. As a tool in data processing, convolution is used for the description of the response of linear shift-invariant systems, and is used in many filter operations. In image processing a filter, or kernel, consists of an image operator changing the value to the image pixels $\mathbf{x}$ depending on the neighborhoods pixel values.

Definition 2.7.6 *Let $f(t)$ and $w(t)$ be two functions. The convolution of f and w, denoted by $w * f$, is the function on $t \geq 0$ given by*

$$g(x) = w * f(x) = \int_0^x w(s) f(x-s)\, ds.$$

This is a particular kind of integral transform:

$$w * f(x) \;=\; \int_{-\infty}^{\infty} w(s)\, f(x-s)\, ds, \tag{2.72}$$

or more generally, if f and w are complex-valued functions on $\mathbb{R}^n$:

$$w * f(x) \;=\; \int_{\mathbb{R}^n} w(s)\, f(x-s)\, ds. \tag{2.73}$$

Since the image is stored as a collection of discrete pixels we need a discrete convolution form. In discrete form the integral (2.73) is replaced by summation, for example for a 2-D function integral (2.73) can be written as:

$$g(x,y) = w(x,y) * f(x,y) \;=\; \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w(s,t)\, f(x-s, y-t)\, ds.$$

A linear spatially invariant filter can be represented with a mask that is convolved with the image array. For a given input 2-D image, size $M \times N$, and a filter mask of size $m \times n$, we can define a neighborhood at point (x,y) by $S_{(x,y)} = \{(x+s, y+t), -a \leq s \leq a, -b \leq t \leq b, \}$, a, b positive integers such that $a = (m-1)/2$ and $b = (n-1)/2$. For example, for a 3 by 3 neighborhood $S_{(x,y)}$, the restriction $f|_{S_{(x,y)}}$ is represented

$$f|_{S_{(x,y)}} = \begin{array}{|c|c|c|} \hline f(x-1,y-1) & f(x-1,y) & f(x-1,y+1) \\ \hline f(x,y-1) & f(x,y) & f(x,y+1) \\ \hline f(x+1,y-1) & f(x+1,y) & f(x+1,y+1) \\ \hline \end{array} \tag{2.74}$$

and also the window mask

$$w = \begin{array}{|c|c|c|} \hline w(-1,-1) & w(-1,0) & w(-1,+1) \\ \hline w(0,-1) & w(0,0) & w(0,+1) \\ \hline w(+1,-1) & w(+1,0) & w(+1,+1) \\ \hline \end{array}.$$

At any point (x,y) the response $g(x,y)$, of the filter is the sum of products of the filter coefficients and the image pixels encompassed by the filter:

$$\begin{aligned} g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + ... \\ +w(0,0)f(x,y) + ... + w(+1,+1)f(x+1,y+1). \end{aligned}$$

A vector representation of linear filtering can be written as the sum of products

$$R = w_1 z_1 + w_2 z_2 + ... + w_{mn} z_{mn} = \sum_{k=1}^{mn} w_k z_k$$

where the w_k are the coefficients of an $m \times n$ filter and the z_k are the corresponding image intensities encompassed by the filter.

We will consider in this section two classes of spacial linear filters: *smoothing linear spacial filters* and *sharpening linear spacial filters.*

Smoothing Linear Spacial Filters. The mean ('average') and Gaussian smoothing filters are commonly used 2-D convolution operators. These operators are used to 'blur' images and remove detail and noise. Small details and noise will be lost in the smoothing process, but sharp edge will become blurry. The mean filter is an easy method of smoothing images which reduces the amount of intensity variation between one pixel and the next by simply replacing each pixel value in an image with the mean ('average') value of its neighbors, including itself. The smoothing spacial mask of a 3×3 is $w = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$. Another important convolution function is the Gaussian function $G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}}$. In practice the Gaussian distribution is effectively zero more than about three standard deviations from the mean, and so we can truncate and get a suitable integer-valued convolution kernel that approximates the Gaussian. A discrete approximation to the Gaussian function with $\sigma = 1$ for 3×3 would be $w = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ or 5x5 is $w = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$. Once a suitable kernel has been calculated, then the Gaussian smoothing can be performed using standard convolution methods. Fig.2.12 shows smoothing with a 5×5 Gaussian kernel of a given image, size 512×512, which has been corrupted by 20% Gaussian noise. The Matlab command used is $g = imfilter(f, fspecial('gaussian', [5, 5], 1))$;.

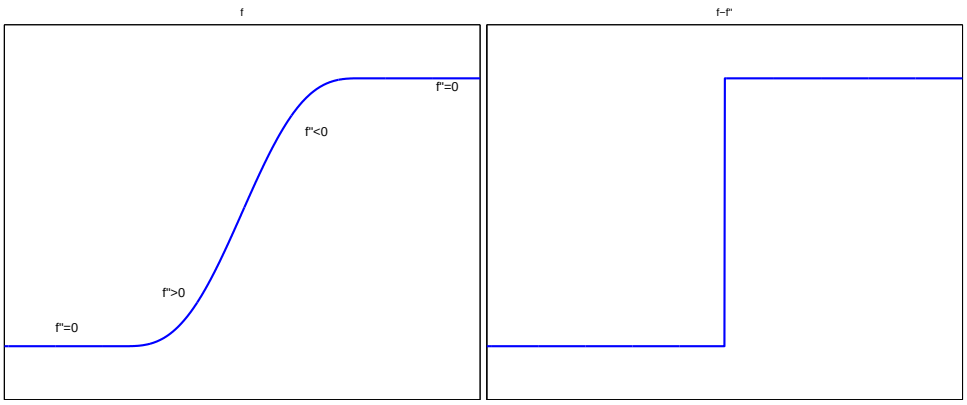
Sharpening Spatial Filters. Sharpening has the opposite effect to smoothing. A blurry image can be made sharper through this process and details like edges will be enhanced.

These filters use the idea of the response of a derivative operator proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied. Intuitively, a digital edge is made up of pixels which are connected and lie on the boundary between two regions. Regions, in this case, refers to two different shades of grey. Assume that a digital grey image function $f(x)$ has a second order partial derivative and the derivatives are defined in terms of differences $\frac{\partial f}{\partial x} = \frac{f(x+h)-f(x-h)}{2h}$

(a) original image (b) noise image (c) smoothed image

Figure 2.12: Gaussian kernel smoothed image with a 5×5 Gaussian kernel.

and $\frac{\partial^2 f}{\partial x^2} = \frac{f(x+h)-2f(x)+f(x-h)}{h^2}$, $h = 1$. Although there are various ways to define these differences, we know that the first derivative (i) must be zero in areas of constant intensity (flat segments); (ii) must be nonzero at the beginning of a gray level step or ramp; (iii) must be nonzero along ramps. Similarly, the second derivative (i) must be zero in every flat segment (constant areas); (ii) must be nonzero at the beginning and end of a gray-level ramp; (iii) must be zero along ramps of constant slope, see Fig. 2.13(a). In the case of Fig. 2.13(b), there is a clear discontinuity detected by the first-and second-order derivatives. By comparing the first and second derivatives we can easily conclude that the first-order derivatives generally detect thicker edges, edges that we can see with our eyes, and the second-order derivatives have a stronger response to finer details, such as thin lines and small points. For this reason, the second-order derivative is better suited for edge detection. For a 1-D function f, shown in figure Fig. 2.13 (a) where the sign of f'' is given, by applying the operation $g = f - f''$ an ideal sharp edge has been expected, as shown in figure Fig. 2.13(b).



(a) x-direction filter (b) x-direction

Figure 2.13: Left: a blurry 1-D edge of the image f. Right: the resulting $g = f - f$" as an ideal sharp image.

Since images are two-dimensional and based on a function $f(x, y)$, we must express edge detection in terms of the gradient and divergence. The gradient of f at spatial

coordinates (x, y) is defined as the vector

$$\nabla f(x,y) = [\frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y)]^T$$

and the magnitude of this vector

$$|\nabla f(x,y)| = \left[(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y}^2)\right]^{1/2}$$

will produce edges. To understand how we can simply recall the basic definition of the derivative and filters. When we take the partial derivative with respect to x, we get $\frac{\partial f}{\partial x} = \frac{1}{2}(1f(x+1,y) - 1f(x-1,y))$ and with respect to y, $\frac{\partial f}{\partial x} = \frac{1}{2}(1f(x,y+1) - 1f(x,y-1))$. Remembering that the pixels of the image under a filter mask is represented as in (2.74) we can create the filter mask in x direction.

$$w_x = \frac{1}{2}\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline -1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}.$$

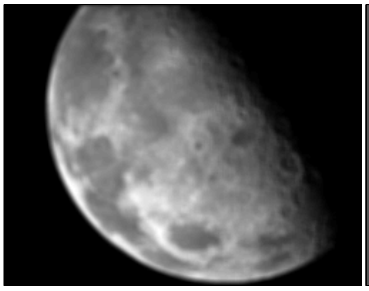
When we filter the image, the mask is essentially taking the partial derivative with respect to x. Also, the rest of the entries in the first and third column of the filter mask can be replaced with -1 and 1 respectively, because those entries are also locations of where the partial derivative with respect to x will be taken. In the same way we will create another filter mask to determine the partial derivative with respect to y. The x-mask is able to detect changes in gray-level intensities in the x-direction (vertical edges), and that the y-mask is changes in the y-direction(horizontal edges) as shown Fig. 2.14(b). Run each mask on separate, but identical images, and then squaring every entry in each of the two resulting filtered images added together, and take the square root of the resulting matrix, gives the magnitude of the gradient vector, shown in Fig. 2.14(c). Another simple isotropic operator is the Laplacian, which, for an image function$f(x, y)$, is defined as

$$\Delta f(x,y) = \frac{\partial^2 f}{\partial x^2}(x,y) + \frac{\partial^2 f}{\partial y^2}(x,y), \tag{2.75}$$

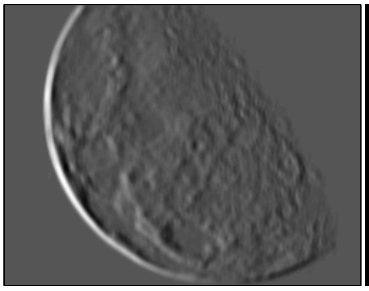
and represents the divergence of derivatives. The discrete Laplacian of two variables, for $h = 1$ as above, is

$$\Delta f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y). \tag{2.76}$$

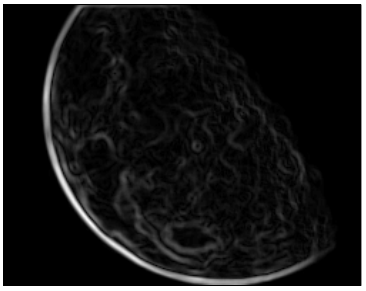
We can notice that the discrete Laplacian can be obtained by using a linear spatial filter with some Laplacian mask $w_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, shown in Fig. 2.14(d). Applying $g(x,y) = f(x,y) - \Delta f(x,y)$ produces an ideal sharp edge, Fig. 2.14(e). Other similar



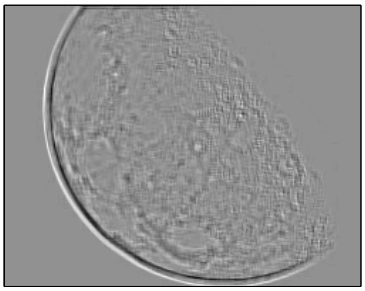
(a) blurry given image



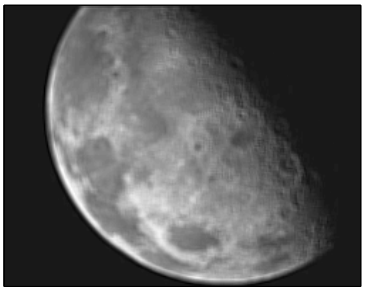
(b) x-direction filter



(c) magnitude of the gradient vector filter



(d) Laplacian filter



(e) enchantment using $g(x,y) = f(x,y) - \Delta f(x,y)$

Figure 2.14: Result of filtering with the x-mask, the magnitude of the gradient and Laplacian filter.

masks can be used.

2.8 Basic Iterative Methods for Solving Linear System of Equations

We begin by setting notation and reviewing some ideas from numerical linear algebra that we are familiar with. Consider the linear system

$$\mathbf{Ax} = \mathbf{b} \tag{2.77}$$

where A is an $m \times n$ matrix, $\mathbf{b}$ is an $m \times 1$ vector and $\mathbf{x}$ is the $n \times 1$ vector of unknowns

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

When the number of equations m is the same as the number of variables n, named a square system of equations, it is possible to have a solution. For this reason in the

following we will consider $m = n$. Finding the unknown $\mathbf{x}$ with *direct methods* would be easy if we know $\mathbf{A}^{-1}$, in case it exists, so that we calculate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \in R^N$. As a numerical technique, *direct methods* (for instance Gaussian elimination) can be used but is rather unusual because they can be computationally very expensive since a lot of memory is needed in real application problems [64]. Introducing a class of *iterative methods* like Jacobi, Gauss-Seidel, SOR, etc. [137, 64, 120, 25] for solving a general linear system of equations can be very useful in terms of implementation and be computationally cheap.

These iterative techniques start with some initial approximation $\mathbf{x}^{(0)}$ and then generate a sequence of vectors $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ which gradually approximates the true solution $\mathbf{x}$ of the linear system via the relation

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}, \tag{2.78}$$

for each $k = 1, 2, 3, \ldots$. In this way the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is converted into an equivalent system of the form $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{c}$ for some fixed matrix $\mathbf{T}$ and a vector $\mathbf{c}$, neitherdepending on time sequence k. Different ways of constructing $\mathbf{T}$ and $\mathbf{c}$ yields different iterative techniques. In the following we will present some of them.

2.8.1 The Jacobi Iterative Method (JAC)

The Jacobi method is one of the simplest iterative methods to implement and at the same time forms the basis of other methods. Given a square system of n linear equations (2.77), if in the ith equation $b_i = \sum_{j=1}^{n} a_{ij}x_j$ we solve for the value of x_i we get

$$x_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} \Big(\frac{-a_{ij}x_j}{a_{ii}}\Big) + \frac{b_i}{a_{ii}}.$$

Then given all the components of $\mathbf{x}^{(k-1)}$ for $k \geq 1$, $x_i^{(k)}$ is generalized by

$$x_i^{(k)} = \frac{1}{a_{ii}}\Big(\sum_{\substack{j=1 \\ j\neq i}}^{n}(-a_{ij}x_j^{(k-1)}) + b_j\Big), \qquad \text{for } i = 1, 2, \ldots, n. \tag{2.79}$$

Let us write (2.79) in the matrix form (2.78). Decompose the matrix $\mathbf{A}$ into a diagonal component $\mathbf{D}$, and the remainder $\mathbf{R}$,

$$\mathbf{D} = \begin{pmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & 0 \\ \vdots & \ldots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n} \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 0 & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & 0 & \cdots & a_{2,n} \\ \vdots & \ldots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & 0 \end{pmatrix},$$

$$\mathbf{A} = \mathbf{D} + \mathbf{R}.$$

The solution is then obtained iteratively via

$$(\mathbf{D} + \mathbf{R})\mathbf{x} = \mathbf{b} \iff \mathbf{D}\mathbf{x} = \mathbf{b} - \mathbf{R}\mathbf{x} \iff \mathbf{x} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x}). \tag{2.80}$$

Here matrix $\mathbf{R}$ can be further decomposed with strictly lower triangular and strictly upper triangular matrices, $\mathbf{L}$ and $\mathbf{U}$ respectively, such that $\mathbf{R} = -\mathbf{L} - \mathbf{U}$, and equation (2.80) can be rewritten:

$$\mathbf{x} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{D}^{-1}\mathbf{b}. \tag{2.81}$$

Thus, comparing with the above representation in (2.78) form, the Jacobi method is

$$\mathbf{x} = \mathbf{T_J}\mathbf{x} + \mathbf{c_J}$$

where $\mathbf{T_J} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$, $\mathbf{c_J} = \mathbf{D}^{-1}\mathbf{b}$.

In practice this method is implemented using equation (2.79) due to the large amounts of memory required in the matrix cases. An advantage of the method is that parallel computation can be implemented, but also that the computation of $x_i^{(k+1)}$ requires each element in $x^{(k)}$ except itself, unlike the Gauss-Seidel method, which overwrite $x_i^{(k)}$ with $x_i^{(k+1)}$, as we will see in the next subsection.

Algorithm 1 Algorithm for Jacobi Method: $x^k \leftarrow JAC(x^{(0)}, A, b, maxit, tol)$.

$k = 1$, $x^{(1)} = x^{(0)}$;
for $iter = 1 : maxit$ **do**
  Compute $x^{(k)}$ using (2.77):
    $x_i^{(k)} \leftarrow \frac{1}{a_{ii}}\Big(\sum_{\substack{j=1 \\ j\neq i}}^{n}(-a_{ij}x_j^{(k-1)}) + b_j\Big)$ for $i = 1, 2, ..., n$
  If $\|x^{(k)} - x^{(k-1)}\| < tol$ or $\|b - Ax^{(k)}\| < tol$ or $iter > maxit$, Break;
  $x^{(k-1)} \leftarrow x^{(k)}$.

end for

2.8.2 Gauss Seidel Method (GS)

The Gauss Seidel method improves on the Jacobi algorithm. The Gauss Seidel method, unlike the Jacobi method, updates a particular point $x_i^{(k)}$ by using the already updated values $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$, which should be better approximations to $x_1, \ldots, x_{i-1}$ than $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$. In this way Gauss Seidel iterations can be written as:

$$x_i^{(k)} = \frac{1}{a_{ii}}\Big(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\Big) \quad \text{for } i = 1, \ldots, n. \tag{2.82}$$

From the above formula we can easily notice that each new entry x_i^k in the GS method has a strong dependence upon all previously evaluated spatial components x_j^k for all $j < i$. This fact makes the order in which the equations are evaluated very important.

Multiplying equation (2.82) by a_{ii}

$$a_{ii}x_i^{(k)} + \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = b_i + \sum_{j=i+1}^{n} \Big(- a_{ij}x_j^{(k-1)} \Big)$$

we get matrix notation for this method,

$$(\mathbf{D} - \mathbf{L})\mathbf{x}^{(k)} = \mathbf{U}\mathbf{x}^{(k-1)} + \mathbf{b}$$

or equivalently

$$\mathbf{x}^{(k)} = \mathbf{T}_{GS}\mathbf{x}^{(k-1)} + \mathbf{c}_{GS} \tag{2.83}$$

where $\mathbf{T}_{GS} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}$ and $\mathbf{c}_{GS} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}$.

Again, the point-wise formula (2.82) is also recommended to save computer memory. One important advantage of the GS method is that with computer implementation there is no need to allocate two arrays $x^{(k-1)}$ and $x^{(k)}$ in the memory, until the updating of $x^{(k)}$ has finished. We can delete every entry $x_i^{(k-1)}$ as soon as it is no longer needed, which means that only one storage vector is required as elements can be overwritten as they are computed, which represents an important saving in computer memory particularly when solving large linear systems.

Algorithm 2 Algorithm for Gauss-Seidel Method: $x^k \leftarrow GS(x^0, A, b, maxit, tol)$.

$k = 1$
for $iter = 1 : maxit$ **do**
  Compute $x^{(k)}$ using (2.82):
    $x_i^{(k)} \leftarrow \frac{1}{a_{ii}}\Big(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\Big)$
  $x_i^{(k-1)} \leftarrow x_i^{(k)}$, for $i = 1, 2, ..., n$
  If $\|b - Ax^{(k)}\| < tol$ or $iter > maxit$, Break;

end for

To speed up the convergence of the method, the equations should be arranged so that a_{ii} is as large as possible.

Equation (2.82) describes the lexicographic order of the grid points which involves ordering the points in increasing order from left to right and bottom to top, starting the update at the bottom left point $(1, 1)$, followed by the approximation at the point $(2, 1)$ and so on, with the top right point $(n - 1, n - 1)$ updated last.

A refinement of the Gauss-Seidel algorithm is Successive Over-Relaxation Method (SOR). The SOR technique can be represented in the form (2.78)

$$\mathbf{x}^{(k)} = \mathbf{T}_{\omega}\mathbf{x}^{(k-1)} + \mathbf{c}_{\omega}, \tag{2.84}$$

where $\mathbf{T}_\omega = (\mathbf{D}-\omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D}+\omega\mathbf{U}]$ and $\omega(\mathbf{D}-\omega\mathbf{L})^{-1}\mathbf{b}$. More detail can be found on [82, 64].

Block iterative methods are generalizations of "point" relaxation schemes, such as Jacobi, Gauss-Seidel, and SOR methods. Fore more see [137].

2.8.3 Convergence

All the iterative methods in the previous subsection define a sequence of iterates of the form

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c},$$

for which $\mathbf{T}$ is a certain iteration matrix. The question addressed in this subsection is: "snake"under which conditions does the iterative method converge, in other words $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{x}^*$, the exact solution. Before introducing the main result of convergence, we present some matrix properties.

Definition 2.8.1 (**Symmetric Matrices**). *A square matrix* $\mathbf{A}$ *is called a symmetric matrix if it is equal to its transpose* $\mathbf{A}^T$*. This is,* $\mathbf{A} = \mathbf{A}^T$*.*

Definition 2.8.2 (**Diagonally Dominant Matrices**). *A matrix* $\mathbf{A}$ *is said to be diagonally dominant if it satisfies*

$$|a_{i,i}| \geq \sum_{j\neq i} |a_{i,j}| \qquad \textit{for all } i,$$

and is called strictly diagonally dominant if

$$|a_{ii}| > \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}|, \qquad \textit{for all } i,$$

where $a_{i,j}$ *denotes the entry in the* i_{th} *row and* j_{th} *column of* $\mathbf{A}$*.*

Definition 2.8.3 (**Positive Definitive Matrices**). *A real symmetric matrix* $\mathbf{A}$ *is positive definite if* $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ *for all non-zero vectors* $\mathbf{x}$*. This is equivalent to say that all the eigenvalues of the matrix are positive.*

Definition 2.8.4 (***Matrix Convergence***). *A square matrix* $\mathbf{A}$ *is said to be convergent if* $\lim_{k\to\infty}\mathbf{A}^k = \mathbf{0}$*.*

A nice consequence of a matrix $\mathbf{A}$ being positive definite is that its inverse $\mathbf{A}^{-1}$ exists. A matrix being diagonally dominant has very nice properties as well. To start, if $\mathbf{A}$ is strictly diagonally dominant then it is non-singular, meaning the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a unique solution. Further, Jacobi and Gauss-Seidel methods for solving a linear system of equations with this type of matrix always converges. Let us now introduce the main result necessary and a sufficient condition for iterative method convergence.

Lemma 2.8.5 *For a given matrix* $\mathbf{A}$*,*

$$||\mathbf{A}|| < 1 \Longleftrightarrow \rho(\mathbf{A}) = \max_i |\lambda_i| < 1,$$

where λ_i are the eigenvalues[5] *of the matrix $\mathbf{A}$, $\rho(\mathbf{A})$ is called the spectral radius of $\mathbf{A}$ and $||.||$ any operator norm ($||\mathbf{A}|| = \max_{v\neq 0} \frac{||\mathbf{Av}||}{||\mathbf{v}||}$).*

Theorem 2.8.6 *If matrix $\mathbf{A} \in \mathbf{R}^n$ then*

$$\lim_{k\to\infty} \mathbf{A}^k = \mathbf{0} \iff \rho(\mathbf{A}) < 1.$$

Theorem 2.8.7 (Spectral Radius and Iterative Convergence). *For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by*

$$\mathbf{x}^{(k)} = \mathbf{T}\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k \geq 1, \tag{2.85}$$

converges to the unique solution of $\mathbf{x}^* = T\mathbf{x}^* + \mathbf{c}$ if and only if $\rho(T) < 1$.

Theorem 2.8.8 *If for a system of equations (2.79), $\mathbf{A}$ is strictly diagonally dominant, the Gauss-Seidel iterations converge for any $\mathbf{x}^{(0)}$.*

Theorem 2.8.9

- *If a matrix $\mathbf{A}$ is strictly diagonally dominant, both Jacobi's and GS converge, and in fact, G-S is faster: $\rho(T_{GS}) \leq \rho(T_J) \leq 1$*
- *If a matrix $\mathbf{A}$ is strictly diagonally dominant, both Jacobi's and G-S converge, and in fact, G-S is faster: $\rho(T_{GS}) < \rho(T_J) < 1$, where T_J and T_{GS} are the iteration matrices for Jacobi and Gauss-Seidel respectively.*

***N*ote**: In a similar way we can prove for Jacobi iterations.

Remark 2.8.10 ***In summary**: We proceed with a review of some of the most popular iterative linear solvers for a linear system $\mathbf{Ax} = \mathbf{b}$, of the form*

$$\mathbf{x}^{(\mathbf{k})} = \mathbf{T}\mathbf{x}^{(k)} + \mathbf{c} \tag{2.86}$$

with different ways of constructing $\mathbf{T}$ and $\mathbf{c}$ yielding different algorithms.

In general, iterative solvers are refinement based methods, and their computational engine is based on residual correction, defined as $\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$ ($\mathbf{r}^k$ is called defect equation for this approximation). For an approximation $\mathbf{x}^k$ of $\mathbf{x}$ and with the assumption that it is possible to obtain an approximation $\bar{A}$ of A in such a way that the system $\bar{A}\mathbf{e} = \mathbf{r}$ is easier to solve than the original system $A\mathbf{x} = \mathbf{b}$, the iterative refinement consists of a three step algorithm:

Algorithm 3 Iterative Method: $x^k \leftarrow IM(x^0, A, b, maxit)$.

for $k = 0$ to *maxit* **do**
  Compute the residual $\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$
  Solve the system $\bar{A}\mathbf{e}^k = \mathbf{r}^k$
  Add the correction $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{e}^k$
end for

[5]For a vector $x \in \mathbb{R}^n$ such that $\mathbf{Ax} = \lambda\mathbf{x}$ for some scalar λ, then lambda is called the eigenvalue of $\mathbf{A}$ with corresponding eigenvector $\mathbf{x}$.

We can certainly notice the similarity to (2.86) with $\mathbf{T} = I - (\bar{A})^{-1}A$. *Since asymptotically we have* $\|\mathbf{x}-\mathbf{x}^{k+1}\| \leq \rho(\mathbf{T})\|\mathbf{x}-\mathbf{x}^{k}\|$ *as* $k \to \infty$, *the speed of convergence of the general iteration (2.86) is characterized by the spectral radius* $\rho(\mathbf{T})$. *Of course if* $\rho(\mathbf{T}) < 1$ *the iterative scheme (2.86) will converge.*

The refinement idea can be used to solve nonlinear systems of equation as well, and this will be shown in the next section.

2.9 Iterative Solution of Nonlinear Equations

We begin by describing our notation. Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ be a nonlinear operator continuously differentiable on $\mathbb{R}^n$ and a nonlinear system of equations usually represented as

$$F(\mathbf{x}) = \mathbf{y},$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of unknowns and $\mathbf{y} = (y_1, \ldots, y_n)$ a fixed vector. We can rewrite the equation in the form:

$$F(\mathbf{x}) = \mathbf{0}, \tag{2.87}$$

with $\mathbf{0}$ representing the zero vector. The basic problem here is to find $\mathbf{x}^* \in \mathbb{R}^n$, a solution of the nonlinear equation (2.87).

In this section, we will start with Newton's method as a refinement idea used to solve an algebraic system of nonlinear equations and will continue with the descent method which has a very desirable property of simplicity of implementation. A special case of the descent method is the time marching scheme. To illustrate the time marching method we bring some implementation used for parabolic equations, such as Explicit and Implicit difference methods and Additive Operator Splitting Scheme (AOS), and discuss their stability. The main references are [73], [51], [120], [119], [131], [173] and [105].

2.9.1 Newton's Method

For a function F as given by (2.87) let J denote the Jacobian matrix $(\partial F_i/\partial x_j)$ of F and assume that J is Lipschitz continuous with constant β in $\mathbb{R}^n$

$$\|J(\mathbf{w}) - J(\mathbf{x})\| \leq \beta\|\mathbf{w} - \mathbf{x}\| \quad \text{for all } \mathbf{w}, \mathbf{x} \in \mathbb{R}^n$$

for any given operator norm. Newton's method attempts to solve the system of nonlinear equations $F(\mathbf{x}) = \mathbf{0}$ by applying the following iterative equation

$$\mathbf{x}^k = \mathbf{x}^{k-1} - J(\mathbf{x}^{k-1})^{-1}F(\mathbf{x}^{k-1}), \qquad k = 1, 2, \ldots. \tag{2.88}$$

In practice, the computing inverse of $J(\mathbf{x}^{k-1})$ may be a difficult task, and is rarely explicitly computed.

$$J(\mathbf{x}^{k-1})(\mathbf{x}^k - \mathbf{x}^{k-1}) = -F(\mathbf{x}^{k-1}) \quad k = 1, 2, \ldots, \tag{2.89}$$

where $\mathbf{x}^0$ is given. We implement (2.89) as refinement

$$\text{Solve } J(\mathbf{x}^{k-1})\mathbf{d}^{k-1} = -F(\mathbf{x}^{k-1})$$

$$\text{Compute } \mathbf{x}^k = \mathbf{x}^{k-1} + \mathbf{d}^{k-1}.$$

Newton's method is attractive because under appropriate conditions it is very quick to converge from a sufficiently good initial guess [73]. In particular, if the Jacobian is nonsingular at the solution, local quadratic convergence can be proved [73] and [120].

Although Newton's method is very quick to converge, requires a very good initial guess; require $J(\mathbf{x}^{k-1})$ at each step; at each step require the solution of a linear system of equation that may be singular or ill-conditioned.

2.9.2 Descent Method

For a nonlinear general functional $F : \mathbb{R}^n \to \mathbb{R}$, the descent method generates a sequence $\mathbf{x}^k$, $k \geq 1$, starting from an initial guess $\mathbf{x}^0 \in \mathbb{R}^n$, using the recurrence

$$\mathbf{x}^k = \mathbf{x}^{k-1} - \alpha^{k-1}\mathbf{d}^{k-1}, \qquad k = 1, 2, \ldots,$$

where the positive step length size α^{k-1} is a scalar allowed to change at each k-iteration and $-\mathbf{d}^{k-1}$ is a search direction given by some formula along which the new iterate $\mathbf{x}^k$ will be chosen. One of the most popular descent methods is ***Gradient descent method*** (or called the steepest descent method). Gradient descent is based on the choice of direction; where F decreases most quickly, which is in the direction opposite to $\nabla F(\mathbf{x}^k)$, supposing F is differentiable at $\mathbf{x}^k$. The result is the scheme

$$\mathbf{x}^k = \mathbf{x}^{k-1} - \alpha^{k-1}\nabla F(\mathbf{x}^{k-1}), \qquad k = 1, 2, \ldots.$$

The main characteristic of descent methods is that the iterates decrease the function value at each stage i.e.

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k). \tag{2.90}$$

Here α^{k-1}, which can be fixed or not, must be selected sufficiently small to satisfy condition (2.90). For α^k equal to *time-step* Δt of a newly introduced time-variable t, the descent method is referred as a ***time-marching*** method.

Time marching methods are in principle the most flexible means of calculating. The basic principle of time marching is to start with a guess and integrate the time dependent equations of motion and energy forward in time until a steady-state solution is obtained. Stability has always been a problem with time marching, limiting the size

of time step and hence increasing the number of steps required to obtain a steady state solution. Many tools of ensuring stability have been devised for the differential form of the equations. The extension from two to three dimensions is relatively simple but slow. Despite this discouragement, time marching seemed to offer the only possibility for solving cases where the currently available methods are thought to be inadequate.

In the following we will give some simple implementation examples of the time marching method for parabolic PDEs of some explicit and implicit schemes. In the following chapters we see how this method can be implemented for solving highly non-linear problems.

2.9.3 Time Marching Schemes for Parabolic PDEs, Explicit Scheme (1-D)

For a given function $u = u(x,t)$ let us consider the 1-D Heat equation

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2}, \qquad 0 \le x \le \pi,\ t \ge 0, \tag{2.91}$$

with a a positive constant, and the boundary conditions

$$\begin{cases} u(x,0) = u_0(x) & (u_0(x) \text{ given}) \\ u(0,t) = 0,\ u(\pi,t) = 0 & \text{for } t > 0. \end{cases}$$

If $a =$*viscosity* we get the so-called Diffusion Equation and if $a =$*thermal conductivity* we get Heat Conduction Equation. The exact solution of this initial value problem (IVP) can be obtained by using the Fourier series method, defining $u_0(x)$ as $-u_0(-x)$ for $-\pi \le x \le 0$, otherwise we use Fourier sine series, but here we use complex Fourier series i.e

$$u(x,t) = \sum_{n=-\infty}^{\infty} A_n e^{\mathbf{i}nx} e^{-n^2 at}, \tag{2.92}$$

where

$$A_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} u_0(x) e^{-\mathbf{i}nx} dx, \qquad \mathbf{i} = \sqrt{-1}.$$

In order to get a numerical approximation to the solution of equation (2.91), we begin by introducing the mesh points. Starting from $t_0 = 0$, $x_0 = 0$ and letting $\triangle x$ and $\triangle t$ be the increments of the variables x and t respectively, with $\triangle x = \frac{\pi}{N}$ and $x_j = j\triangle x$ and $t_k = k\triangle t$ where $N \in Z$ and $j = 1, 2, \ldots, N$, $k = 0, 1, \ldots$. Also let $u_j^k = u(x_j, t_k)$, be the value of the function at grid point (x_j, t_k). Replacing the derivatives in equation (2.91) by their finite difference approximations, and rearranging terms, we end up with the linear system:

$$u_j^{k+1} = u_j^k + \frac{a\triangle t}{(\triangle x)^2}\Big(u_{j+1}^k - 2u_j^k + u_{j-1}^k\Big), \tag{2.93}$$

where $j = 1, 2, \ldots, N-1$, $k = 0, 1, \ldots$. The boundary conditions will be

$$u_0^k = 0, \quad u_N^k = 0, \quad k = 0, 1, \ldots, \tag{2.94}$$

and the initial conditions

$$u_j^0 = u_0(j\triangle x), \quad j = 0, 1 \ldots, N. \tag{2.95}$$

These equations can be used recursively to determine u_j^{k+1} for $0 \leq j \leq N$ and $k \geq 0$. The explicit nature of the difference method implies that the approximate solution is given in the following matrix form:

$$u_j^{k+1} = \mathbf{A} u_j^k, \tag{2.96}$$

where A is a $(N-1)$ by $(N-1)$ matrix

$$\mathbf{A} = \begin{bmatrix} (1 - 2\frac{a\triangle t}{(\triangle x)^2}) & \frac{a\triangle t}{(\triangle x)^2} & 0 & \cdots & 0 \\ \frac{a\triangle t}{(\triangle x)^2} & (1 - 2\frac{a\triangle t}{(\triangle x)^2}) & \frac{a\triangle t}{(\triangle x)^2} & \cdots & 0 \\ 0 & \frac{a\triangle t}{(\triangle x)^2} & (1 - 2\frac{a\triangle t}{(\triangle x)^2}) & \cdots & 0 \\ \vdots & \ddots & \ddots & \ldots & \vdots \\ 0 & 0 & 0 & \frac{a\triangle t}{(\triangle x)^2} & (1 - 2\frac{a\triangle t}{(\triangle x)^2}) \end{bmatrix}.$$

Stability: Stability can be checked using Fourier or Von Neumann analysis. Substituting $u_j^k = A_n G^k e^{\mathbf{i} n j \triangle x}$, $n \in Z$ in equation (2.93), with A_n the same as that for the exact solution (2.92) one can find the stability condition for the explicit schemes

$$\frac{2a\triangle t}{(\triangle x)^2} \leq 1 \quad \overset{\text{since } a>0}{\Rightarrow} \quad \triangle t \leq \frac{(\triangle x)^2}{2a}.$$

The Fourier or Von Neumann analysis is not only applied to simple problems such as constant coefficients here, but can be applied to differential equations with variable coefficients. This method can be applied to a wide variety of difference equations, where the more elementary methods fail. If the above condition is not satisfied, then the symptom of instability shows up in a relatively small number of iterations. For instance a simple example of the above condition, if $a = 1$, and the space mesh size $x = .01$, then we must take time step size $t \leq .00005$. Since not all choices of space and time steps lead to a convergent scheme, the explicit scheme (2.93) is called conditionally stable. An unconditionally stable method, one that does not restrict the time step, can be constructed by applying the backwards difference formula to equation (2.91) as we will see in the following.

2.9.4 Time Marching Schemes for Parabolic PDEs, Implicit schemes (1-D)

A fully implicit method for equation (2.91) computes $\frac{\partial^2 u}{\partial x^2}$ at the new time $(k+1)\triangle t$. Replacing the derivatives in equation (2.91) with the backwards difference formula to compute $\frac{\partial^2 u}{\partial x^2}$ we have

$$\frac{u_j^{k+1} - u_j^k}{\triangle t} = a\frac{u_{j+1}^{k+1} - 2u_j^{k+1} + u_{j-1}^{k+1}}{(\triangle x)^2}, \tag{2.97}$$

where $j = 1, 2, \ldots, N-1\,, k = 0, 1, \ldots$.

In general the implicit difference system for the 1-D heat equation (2.91) has the form:

$$\frac{u_j^{k+1} - u_j^k}{\triangle t} = a\frac{\theta(\delta^2 u)_j^{k+1} + (1-\theta)(\delta^2 u)_j^k}{(\triangle x)^2} \tag{2.98}$$

where δu_j and $\delta^2 u_j$ are the central differences calculated as

$$\delta u_j = u((j+\frac{1}{2})\triangle x) - u((j-\frac{1}{2})\triangle x);\;\; \delta^2 u_j = u((j+1)\triangle x) - 2u(j\triangle x) + u((j-1)\triangle x).$$

with j an integer or an integer plus $\frac{1}{2}$, and $0 \le \theta \le 1$ is a real constant (when $\theta = 0$, the system reduces to (2.93)).

Stability: It can be shown that the stability condition for in general form (2.98) must be

$$\begin{cases} \frac{2a\triangle t}{(\triangle x)^2} \le \frac{1}{1-2\theta} & \text{if } 0 \le \theta < 1/2 \\ \text{always stable} & \text{if } 1/2 \le \theta \le 1 \end{cases}$$

or

$$\begin{cases} \triangle t \le \frac{(\triangle x)^2}{2a(1-2\theta)} & \text{if } 0 \le \theta < 1/2 \\ \text{always stable} & \text{if } 1/2 \le \theta \le 1. \end{cases}$$

In the next section we discuss the 2-D case of the heat equation.

2.9.5 Time Marching Schemes for Parabolic PDEs in 2-D

The natural generalization of the one-dimensional problem (2.91) in two dimensions is the equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= div(b\nabla u)\;\; (b > 0) \\ &= b\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}, \end{aligned} \tag{2.99}$$

where $u = u(x, y, t)$, $x \in [a, b], y \in [c, d]$. Suppose that the initial condition is given and the function u satisfies boundary conditions in both x- and $y-$directions. As before, we discretize in time $t_n = t_0 + k\triangle t,\;\; k = 0, 1, 2, ...$ and for both the x- and y-directions, we use the uniform grid $x_j = x_0 + j\triangle x$, $y_l = y_0 + l\triangle y$, with $\triangle x = \frac{b-a}{M}, \triangle y = \frac{d-c}{N}$. Also

let $u_{jl}^k = u(x_j, y_l, t_k)$, be the value of the function at grid point (x_j, y_l, t_k).

$$u_{jl}^k = u(j\triangle x, l\triangle y, k\triangle t)$$

The finite difference approximation for (2.99), can be written as follows:

$$\frac{u_{jl}^{k+1} - u_{jl}^k}{\triangle t} = \theta\phi_{jl}^{k+1} + (1-\theta)\phi_{jl}^k, \tag{2.100}$$

where ϕ_{jl}^k is a central two spatial dimensions approximation scheme for $b\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}$:

$$\phi_{jl}^k = \frac{b}{(\triangle x)^2}\Big(u_{j+1,l}^k - 2u_{jl}^k + u_{j-1,l}^k\Big) + \frac{b}{(\triangle y)^2}\Big(u_{j,l+1}^k - 2u_{jl}^k + u_{j,l-1}^k\Big). \tag{2.101}$$

If $0 < \theta < 1$ then (2.100) is a semi-implicit, if $\theta = 0$, then it reduces to explicit and if $\theta = 1$ it reduces to implicit.

Stability: For a parabolic 2-D PDE one have the following following stability condition

$$\begin{cases} \frac{b\triangle t}{(\triangle x)^2} + \frac{b\triangle t}{(\triangle y)^2} \leq \frac{1}{2-4\theta} & \text{if } 0 \leq \theta < 1/2 \\ \text{always stable} & \text{if } 1/2 \leq \theta \leq 1. \end{cases}$$

2.9.6 Additive Operator Splitting (AOS) Scheme

Performing an explicit scheme on a nonlinear diffusion equation brings stability problems and the requirement of very small time steps, which leads to poor efficiency and limited practical use. Introduced by Tai [106] in 1992 and Weickert [175] in 1998 the AOS scheme guarantees equal treatment of all coordinate axes and is stable for big time steps. The scheme presents a semi-implicit algorithm based on a discrete nonlinear diffusion scale-space framework. This scheme applies to the diffusion equation given in the following form

$$u_t = div(g\nabla u) + \mathrm{f}(\mathbf{x}, u) \tag{2.102}$$

$$= (gu_{x_1})_{x_1} + \ldots + (gu_{x_m})_{x_m} + \mathrm{f}(\mathbf{x}, u), \tag{2.103}$$

in $[0,T] \times \Omega \subset \mathbb{R}^m$, and with initial and boundary conditions

$$u(0,.) = u_0 \quad and \quad \frac{\partial u}{\partial n} = 0 \quad on \ \partial\Omega,$$

where n denotes the normal to the boundaries $\partial\Omega$. Here g is the diffusivity function and f denotes a reaction term. For example, in image segmentation g is chosen as a nonlinear diffusion filter, such as the edge detector filter function $g(|\nabla u|)$, with the property $0 \leq g(|\nabla u|) \leq 1$. The nonlinear diffusion filter belongs to a much larger filter class but we consider this filter since we will mainly deal with it in what follows. Most implementations of nonlinear diffusion filters are based on finite difference methods, and the scalar-valued diffusivity g can be replaced by a smooth matrix-valued function

that remains uniformly positive definite as long as its argument is bounded. For such a matrix some properties such as well-posedness, average grey value invariance, causality in terms of an extremum principle and Lyapunov functionals, and convergence to a constant steady-state can be established [169, 170]. The requirement needed are easy to check for many discretizations, as we show in the following.

1-Dimensional Scheme

The 1-D diffusion equation as given in (2.102) can be written

$$\partial_t u = \partial_x(g\partial_x u) + \mathrm{f}(\mathbf{x}, u) \tag{2.104}$$

where u is given in a bounded domain $\Omega = [a, b]$. Denoting x_i the discrete elements of the segment $[a, b]$, divided in N equidistant subintervals with distance $h = \frac{b-a}{N-1}$, such that $x_i = a + ih$, and discrete time $t_k = k\Delta t$, $k = 1, 2, ... \in \mathbb{N}_0$, with Δt time step, the approximation of the function u to the point x_i in a discrete time t_k can be estimated as $u_i^k = u(x_i, t_k)$. A backward Euler implicit step for the time discretization and a spatial finite difference discretization yields the semi-implicit evolution scheme on (2.104), reflecting the boundaries condition, is given by

$$\begin{aligned}
&\frac{u_i^{k+1} - u_i^k}{\Delta t} = \frac{g^k \partial_x u^{k+1}|_{i+1/2} - g^k \partial_x u^{k+1}|_{i-1/2}}{h} + f(u_i^k) \\
\Rightarrow &\frac{u_i^{k+1} - u_i^k}{\Delta t} = \frac{\frac{g_{i+1}^k + g_i^k}{2}\frac{u_{i+1}^{k+1} - u_i^{k+1}}{h} - \frac{g_{i-1}^k + g_i^k}{2}\frac{u_i^{k+1} - u_{i-1}^{k+1}}{h}}{h} + f(u_i^k), \\
&\Rightarrow \frac{u_i^{k+1} - u_i^k}{\Delta t} = \sum_{j \in \mathcal{N}(i)} \frac{g_j^k + g_i^k}{2h^2}(u_j^{k+1} - u_i^{k+1}) + f(u_i^k),
\end{aligned} \tag{2.105}$$

where $\mathcal{N}(i)$ is the set of the two neighbours of element i. Equation (2.105) can be rewrite in matrix form

$$(I - \Delta t A(u^k))u^{k+1} = u^k + \Delta t f(u^k),$$

with $A(u^k) = a_{ij}^k$ and

$$a_{ij}^k = \begin{cases} \frac{g_j^k + g_i^k}{2h^2} & j \in \mathcal{N}(i) \\ -\sum\limits_{n \in \mathcal{N}(i)} \frac{g_j^k + g_n^k}{2h^2} & j = i \\ 0 & else \end{cases},$$

where $I \in \mathbb{R}^N$ is the unit matrix. Without loss of generality we drop the term f and we come down to the iteration scheme

$$(I - \Delta t A(u^k))u^{k+1} = u^k.$$

This is a semi-implicit scheme and to calculate the u^{k+1} it is required to solve the system of equation

$$u^{k+1} = \Big(I - \Delta t A(u^k)\Big)^{-1} u^k, \qquad k = 1, 2, \ldots. \tag{2.106}$$

To show the reliability of these schemes it is required to verify that they satisfy established criteria for nonlinear diffusion scale-spaces [169, 170] for arbitrarily large time steps. In particular, this makes the scheme unconditionally stable and it will not suffer from any time step size restriction. For this reason the scheme can be fully adapted to the desired accuracy without the need to choose small time steps for stability reasons.

For a given discrete scheme of type

$$\begin{aligned} u^0 &= f \\ u^{k+1} &= Q(u^k)u^k, \quad \forall k \in \mathbb{N}_0 \end{aligned}$$

all one has to check is that the following criteria holds:

D1) Continuity in Its Argument:

$$Q \in C(\mathbb{R}^N, \mathbb{R}^N \times \mathbb{R}^N) \tag{2.107}$$

D2) Symmetry:

$$q_{ij} = q_{ji}, \quad j, i \in J \tag{2.108}$$

D3) Unit Row Sum:

$$\sum_{j \in J} q_{ij} = 1, \quad \forall i \in J \tag{2.109}$$

D4) Nonnegativity:

$$q_{ij} \geq 0, \quad \forall j, i \in J \tag{2.110}$$

D5) Positive Diagonal:

$$q_{ii} \geq 0, \quad \forall i \in J \tag{2.111}$$

D6) Irreducibility: Any two elements can be connected by a path with nonvanishing diffusivities. Formally: For $\forall j, i \in J$ there exist $k_0, \ldots, k_r \in J$ with $k_0 = i$, and $k_r = j$ such that $q_{k_p k_{p+1}} \neq 0$ for $p = 0, \ldots, r-1$.

We can check D1–D6 and show that the semi-implicit scheme (2.106) establishes a discrete scale-space.

D1) It can be shown that $B(u^k) = [b_{ij}(u^k)] = I - \Delta t A(u^k)$ is invertible, since it is strictly diagonally dominant $b_{ii} > \sum_{i \neq j} b_{ij}, \ \forall i \in J$. Thus, $Q(u^k) = [B(u^k)]^{-1}$ exists and

the continuity of Q in its argument follows from the continuity of g.

D2) The symmetry of A also carries over Q.

D3) Since B has unit row sum, we have $Bw = w$, for $w = (1, 1, \ldots, 1)^T \in \mathbb{R}^N$. Thus $w = B^{-1}w = Qw$.

D4–D6) We already know that B is strictly diagonally dominant. It is also immediately seen that B is irreducible, $b_{ij} \leq 0$ for $i \neq j$ and $b_{ii} > 0$ for all i. Then a theorem by Varga [163] tells us that $Q = B^{-1}$ satisfies

$$q_{ij} > 0, \quad \forall i, j \in J \tag{2.112}$$

Thus, Q is nonnegative, has positive diagonal and is irreducible.

Given a system resulting from implicit schemes with three space points $i-1; i; i+1$, so called tridiagonal linear system, can be solved using the Thomas algorithm.

m-Dimensional Scheme

Recalling the m-dimensional diffusion equation (2.102) given in the form

$$\frac{\partial u}{\partial t} = \sum_{j=1}^{m} \frac{\partial}{\partial x_j}(g_j(u)\frac{\partial u}{\partial x_j}) + \mathrm{f}(\mathbf{x}, u) \tag{2.113}$$

in $[0, T] \times \Omega \subset \mathbb{R}^m$, with initial and boundary conditions

$$u(0, .) = u_0 \quad and \quad \frac{\partial u}{\partial n} = 0 \quad on \ \partial\Omega,$$

we would like to have an unconditionally stable method like the one introduced in 1-D. A discrete m-dimensional function u can be regarded as a vector with element i representing the location x_i. Let h_l denote the grid size in the l direction. We consider discrete times $t_k := k\Delta t$, where $k \in \mathbb{N}_0$ and Δt is the time step size. By u_i^k and g_i^k we denote approximations to $u(x_i, t_k)$ and $g(u(x_i, t_k))$. With a backward Euler implicit step for the time discretization and a spatial finite difference in the same manner as in the 1-D case, a semi-implicit discretization of the diffusion equation reflecting the boundary conditions is given by

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \sum_{l=1}^{m} \sum_{j \in \mathcal{N}_l(i)} \frac{g_j^k + g_i^k}{2h^2}(u_j^{k+1} - u_i^{k+1}) + f(u_i^k), \tag{2.114}$$

where $\mathcal{N}_l(i)$ consists of the two neighbours of the element i in the l direction (boundary elements may have only one neighbour). In this vector–matrix notation we can write the m-dimensional implicit scheme as

$$\frac{u^{k+1} - u^k}{\Delta t} = \sum_{l=1}^{m} A_l(u)u^{k+1} + f(u^k)$$

or

$$(I - \Delta t \sum_{l=1}^{m} A_l(u)) u^{k+1} = u^k + \Delta t f(u^k).$$

Without loss of generality we drop the reaction term f and we have

$$(I - \Delta t \sum_{l=1}^{m} A_l(u)) u^{k+1} = u^k$$

$$u^{k+1} = \Big(I - \Delta t \sum_{l=1}^{m} A_l(u)\Big)^{-1} u^k, \qquad k = 1, 2, \ldots, \tag{2.115}$$

where u^k is represented by a column vector of length N^m. For dimensions $m \geq 2$, the matrix may reveal a much larger bandwidth for A. Applying direct algorithms such as Gaussian elimination would destroy the zeros within the band and would lead to an immense storage and computation effort [175]. Classical iterative algorithms become slow for large Δt, since this increases the condition number of the system matrix. In the following we introduce a splitting-based alternative method where the one-dimensional *Thomas algorithm* can be used [175]. The scheme proposes

$$u^{k+1} = \frac{1}{m} \sum_{l=1}^{m} \Big(I - m \triangle t A_l\Big)^{-1} u^k, \qquad k = 1, 2, \ldots, \tag{2.116}$$

where the operators $B_l(u^k) := I - m \triangle t A_l(u^k)$ describe one-dimensional diffusion processes along the x_l axes. It can be noticed that each iteration step requires the old iterate to be propagated in each coordinate direction separately. Then the new iteration is given by the average of these intermediate solutions and is well-suited for parallel computing. It is stated without proof in [175] that the AOS scheme is an $\mathcal{O}(\Delta t) + \mathcal{O}(h^2)$ accurate finite difference approximation to the original equation, and a proof is given in [119].

Theorem 2.9.1 *The AOS scheme (2.116) with m=2 corresponding to the finite difference equation*

$$(\mathcal{F}u)^k := \frac{1}{\Delta t} u^{k+1} - \frac{1}{2\Delta t}\Big(I - 2\frac{\Delta t}{h^2} A_1\Big)^{-1} u^k - \frac{1}{2\Delta t}\Big(I - 2\frac{\Delta t}{h^2} A_2\Big)^{-1} u^k = 0,$$

$k = 0, 1, \ldots,$ is consistent in l_∞-norm of first order in time and second order in space with the PDE (2.113).

Proof Let u be an arbitrary smooth function, for which we will examine the local truncation error

$$T_{ij}^k(u) = (\mathcal{F}u)_{ij}^k - L(u)(t_k, x_{ij}),$$

where L is the spatial partial differential operator given by

$$L(u) = div(g \nabla u) = (g u_x)_x + (g u_y)_y.$$

Besides the function u, we will need in each time step the function ν solving

$$\nu_t - 2(g\nu_x)_x = 0 \qquad \text{with initial condition} \quad \nu(t_k, .) = u(t_k, .) \tag{2.117}$$

and the function ω solving

$$\omega_t - 2(g\omega_y)_y = 0 \qquad \text{with initial condition} \quad \omega(t_k, .) = u(t_k, .), \tag{2.118}$$

supplied with usual Neumann boundary conditions. These equations are one-dimensional diffusion equations, which are accelerated by the factor 2 compared to the PDE (2.113). Let r be the function such that

$$u = \frac{1}{2}\nu + \frac{1}{2}\omega + r \qquad in\,[t_k, t_{k+1}] \times \Omega$$

at each time step k. For the finite difference operator at time step k, we obtain

$$\begin{aligned}(\mathcal{F}u)^k = &\underbrace{\frac{1}{2\Delta t}\mathbf{v}^{k+1} - \frac{1}{2\Delta t}\Big(I - 2\frac{\Delta t}{h^2}A_1\Big)^{-1}u^k}_{F_1} \\ &+ \underbrace{\frac{1}{2\Delta t}\mathbf{w}^{k+1} - \frac{1}{2\Delta t}\Big(I - 2\frac{\Delta t}{h^2}A_2\Big)^{-1}u^k}_{F_2} + \frac{1}{\Delta t}\mathbf{r}^{k+1},\end{aligned}$$

where u^k, u^{k+1}, $\mathbf{v}^{k+1}$, $\mathbf{w}^{k+1}$, $\mathbf{r}^{k+1}$denote the vectors (reordered already by rows) containing the values u, v and w on the spatial grid point at time steps k and $k+1$ respectively. Consider F_1

$$\frac{1}{2\triangle t}\mathbf{v}^{k+1} - \frac{1}{2\triangle t}\Big(I - 2\frac{\triangle t}{h^2}A_1\Big)^{-1}u^k$$

which apart from the factor $\frac{1}{2}$ corresponds to the very first step of an implicit scheme for the PDE (2.117). Since ν is initialized to $u(t_k, .)$ at time step k, this term is nothing else but the finite difference operator of the implicit scheme evaluated at the function ν. But ν is also the solution of this PDE with initial data u^k, so that the local truncation error for the implicit scheme is of first order in time and second order in space i.e

$$\Big\|\frac{1}{2\Delta t}\mathbf{v}^{k+1} - \frac{1}{2\Delta t}\Big(I - 2\frac{\Delta t}{h^2}A_1\Big)^{-1}\mathbf{u}^k\Big\|_\infty = \mathcal{O}(\Delta t) + \mathcal{O}(h^2).$$

The same also applies to F_2, using the fact that ω solves the PDE (2.118). So we obtain

$$(\mathcal{F}u)_{ij}^k = \frac{1}{\Delta t}\mathbf{r}_{ij}^{k+1} + \mathcal{O}(\Delta t) + \mathcal{O}(h^2), \tag{2.119}$$

where the constants in the order terms do not depend on i, j or k. It remains to take a look at $\mathbf{r}_{ij}^{k+1}$:

$$\mathbf{r}_{ij}^{k+1} = \mathbf{u}_{ij}^{k+1} - \frac{1}{2}\mathbf{v}_{ij}^{k+1} - \frac{1}{2}\mathbf{w}_{ij}^{k+1}.$$

Using the Taylor expansion of u, ν and ω at time k we get

$$\begin{aligned}
\mathbf{r}_{ij}^{k+1} &= u_{ij}^k + \Delta t u_t(t_k, x_{ij}) + \mathcal{O}((\Delta t)^2) \\
&\quad - \frac{1}{2}(v_{ij}^k + \Delta t \nu_t(t_k, x_{ij}) + \mathcal{O}((\Delta t)^2) \\
&\quad - \frac{1}{2}(w_{ij}^k + \Delta t \omega_t(t_k, x_{ij}) + \mathcal{O}((\Delta t)^2) \\
&= \Delta t\Big(u_t(t_k, x_{ij}) - \frac{1}{2}\nu_t(t_k, x_{ij}) - \frac{1}{2}\omega_t(t_k, x_{ij})\Big) + \mathcal{O}((\Delta t)^2),
\end{aligned}$$

substituting into equation (2.119) and the fact that ν and ω satisfy diffusion equations at time t_k, so $u = \nu = \omega$, hence, $(gu_x)_x = (g\nu_x)_x$ and $(gu_y)_y = (g\omega_y)_y$ and we obtain

$$\begin{aligned}
(\mathcal{F}u)_{ij}^k &= u_t(t_k, x_{ij}) - \frac{1}{2}2(u\phi_x)_x(t_k, x_{ij}) \\
&\quad - \frac{1}{2}2(gu_y)_y(t_k, x_{ij}) + \mathcal{O}(\Delta t) + \mathcal{O}(h^2) \\
&= u_t(t_k, x_{ij}) - (div(gu))(t_k, x_{ij}) + \mathcal{O}(\Delta t) + \mathcal{O}(h^2).
\end{aligned}$$

Putting this into the local truncation error yields the required consistency

$$T_{ij}^k(u) = \mathcal{O}(\Delta t) + \mathcal{O}(h^2).$$

The corresponding assertion for the l_∞-norm follows from the fact that the constants in the order terms do not depend on time or location [119]. □

In summary, the explicit methods require less computation per step and storage effort is linear in the number of pixels N, but are only conditionally stable and so may require far smaller step sizes than an implicit method of nominally the same order. For explicit scheme we have limitations to its stability of the form $\Delta t < \frac{(\Delta x)^2}{2a}$, which is often a very severe step size restriction. It means that we can use the implicit schemes, which are not limited by their stability, for the same accuracy as the explicit one. In one dimension, the semi-implicit scheme requires solving a diagonally dominant tridiagonal system of equations; this is unconditionally stable and we can get a desired accuracy without the need to choose small time steps for stability. In higher dimensions the semi-implicit scheme remains absolutely stable but it is laborious to solve the system of equations obtained from the discretization. AOS method splits the m-dimensional spatial operator into the sum of m one dimensional space discretizations. The update of each grid point involves only two neighbours in each dimension, thus reducing the system to a set of tridiagonal systems, which is equivalent to a semi-implicit scheme in one dimension. In practice, it does not make much sense to use extremely large time steps for this method and the implicit one, since this leads to poor rotation invariance, as splitting effects become visible. In practice all one has to do is to replace the explicit scheme by an AOS scheme with 10 or 20 times larger time step sizes.

Improving the time marching or AOS in an effective way can be done by using a high order accuracy method. A relatively new method, namely homotopy analysis method [94], attracts with a simple way of approximating the PDEs with a higher accuracy. The method transforms the nonlinear problems to solution series of linear problems.

In the next Section we show that this method has the potential to be used for different linear and nonlinear differential equations.

2.9.7 Homotopy Analysis Method

In this section, we briefly introduce the standard homotopy analysis method (HAM) for a general nonlinear problem. This will be the first step to apply the DHAM (as a discrete form of HAM). This method has been known as an analytic method for solving nonlinear problems [97]. The method attracts with a simple way of controlling and adjusting the convergence region and rate of solution series of nonlinear problems. This method has been improved [99, 100, 95, 101, 97, 98] and widely used to solve linear or nonlinear ODE and PDE problems showing great performance. Different fields in science, engineering, finance or technological fields have employed the method to solve or improve many types of problem [184, 1, 72, 79, 50, 2, 139] or even to find some new solutions of a few nonlinear equations which have never been solved by previous analytic methods or even numerical methods [96]. Unlike the perturbation techniques method [116, 117] or the so-called non-perturbation techniques, such as the δ-expansion method [7], Adomian's decomposition method [5] and so on, HAM does not depend on small/large physical parameters and is valid not only for weakly nonlinear problems, but also for strongly nonlinear problems and at the same time the convergence of the solution has been proved [97]. The method transforms the nonlinear problem to a hight order linear approximation.

Consider a non-linear equation of the following form:

$$\mathcal{N}[u(\mathbf{x},t)] = 0 \tag{2.120}$$

subject to the initial condition

$$u(\mathbf{x},0) = u_0(\mathbf{x}) \tag{2.121}$$

where $\mathcal{N}$ is a nonlinear operator which represents the whole equation, $\mathbf{x} \in R^n$, t denotes independent variables and $u(\mathbf{x})$ is an unknown function. Based on Liao [99], the following zero-order deformation equation is constructed from the original equation (2.120), as follows:

$$(1-q)\mathcal{L}[\varphi(\mathbf{x},t;q) - u_0(\mathbf{x},t)] = q\hbar\mathcal{H}(\mathbf{x},t)\mathcal{N}[\varphi(\mathbf{x},t;q)] \tag{2.122}$$

where $u_0(\mathbf{x},t)$ is an initial guess, $\hbar$ is an auxiliary parameter, $q \in [0,1]$ is an embedding parameter, $\varphi(\mathbf{x},t;q)$ is a function of t and q, $\mathcal{H}$ is a nonzero auxiliary function and $\mathcal{L}$

is an auxiliary linear operator with the following property

$$\mathcal{L}[\varphi(\mathbf{x},t)] = 0 \quad \text{when} \quad \varphi(\mathbf{x},t) = 0. \tag{2.123}$$

It should be emphasized that we have the freedom to choose the initial approximation, the auxiliary linear operator $\mathcal{L}$, the auxiliary parameter $\hbar$ and the auxiliary function $\mathcal{H}$. Obviously, since $\hbar \neq 0$, $\mathcal{H} \neq 0$, when $q = 0$ and $q = 1$, it holds that $\varphi(\mathbf{x},t;0) = u_0(\mathbf{x},t)$ and $\varphi(\mathbf{x},t;1) = u(\mathbf{x}),t$, respectively. Thus, as q increases from 0 to 1, the solution $\varphi(\mathbf{x},t;q)$ deforms from the initial guess $u_0(\mathbf{x},t)$ to the solution $u(\mathbf{x},t)$. Expanding $\varphi(\mathbf{x},t;q)$ in the Taylor series with respect to q, one has

$$\varphi(\mathbf{x},t;q) = u_0(\mathbf{x},t) + \sum_{m=1}^{\infty} u_m(\mathbf{x},t)q^m, \tag{2.124}$$

where

$$u_m(\mathbf{x},t) = \frac{1}{m!}\frac{\partial \varphi^m(\mathbf{x},t;q)}{\partial q^m}\bigg|_{q=0}. \tag{2.125}$$

If the auxiliary linear operator, the initial guess, the auxiliary parameter $\hbar$, and the auxiliary function $\mathcal{H} \neq 0$ are chosen such that the series (2.124) converges at $q = 1$, one has

$$u(\mathbf{x},t) = u_0(\mathbf{x},t) + \sum_{m=1}^{\infty} u_m(\mathbf{x},t), \tag{2.126}$$

which must be one of the solutions of the original nonlinear equation, as proved by Liao [94].

Let us define $\vec{u}_k(\mathbf{x}) = \{u_0(\mathbf{x}), \ldots, u_k(\mathbf{x})\}$ as the vector of the composed solutions. According to the definition for $u_m(\mathbf{x},t)$ (2.125), the governing equation and the corresponding initial condition of $u_m(\mathbf{x},t)$ can be deduced from the zero-order deformation equation (2.122) and (2.123) as follows. Differentiating the zero-order deformation equation (2.122) m-times with respect to q and dividing them by $m!$ and finally setting $q = 0$, we obtain the following m^{th}-order deformation problem:

$$\mathcal{L}[u_m(\mathbf{x},t) - \chi_m u_{m-1}(\mathbf{x},t)] = \hbar\mathcal{H}R_m[\vec{u}_{m-1}(\mathbf{x},t)] \tag{2.127}$$

where

$$R_m[\vec{u}_{m-1}(\mathbf{x},t)] = \frac{1}{(m-1)!}\frac{\partial^{m-1}\mathcal{N}[\varphi(\mathbf{x},t;q)]}{\partial q^{m-1}}\bigg|_{q=0} \tag{2.128}$$

and

$$\chi_m = \begin{cases} 1, & m > 1 \\ 0, & m \leqslant 1 \end{cases}$$

Applying the inverse operator $\mathcal{L}^{-1}$ on both sides of equation (2.127) we can obtain

$$u_m(\mathbf{x},t) = \chi_m u_{m-1}(\mathbf{x}) + \mathcal{L}^{-1}\{\hbar\mathcal{H}R_m[\vec{u}_{m-1}(\mathbf{x},t)]\} \tag{2.129}$$

and at the Mth order we have $u_M(\mathbf{x},t) = \sum_{m=0}^{M} u_m(\mathbf{x},t)$. Apparently as M increases we

may obtain a more accurate approximate solution of the original equation (2.120). Note that when solving the mth-order deformation equations (2.127) $R_m[\vec{u}_{m-1}(\mathbf{x})]$ depends only on $u_0, u_1, u_2, ..., u_{m-1}$, which are already known. In our applications we will use two different linear operators, as follows[6]:

$$\mathcal{L}_1(\varphi(\mathbf{x},t;q)) = \frac{\partial\varphi(\mathbf{x},t;q)}{\partial t} + \theta\varphi(\mathbf{x},t;q), \tag{2.130}$$

and

$$\mathcal{L}_2(\varphi(\mathbf{x},t;q)) = (1+t)\frac{\partial\varphi(\mathbf{x},t;q)}{\partial t} + \varphi(\mathbf{x},t;q), \tag{2.131}$$

with the properties $\mathcal{L}_1[C_1e^{-t}] = 0$, $\mathcal{L}_2[\frac{C_2}{1+t}] = 0$ and θ is a positive constant. It can be easy found that $\mathcal{L}_1^{-1} = e^{-\theta t}\int_0^t e^{\theta\tau}\varphi d\tau$ and $\mathcal{L}_2^{-1} = \frac{1}{1+t}\int_0^t \varphi d\tau$.

Below we shall write u^{DHAM1} and u^{DHAM2}, notation used to make the distinction between the linear operators $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively, for finding the solution $u(\mathbf{x})$ of the equation (2.120).
By applying the inverse operators $\mathcal{L}_1^{-1}$ and $\mathcal{L}_2^{-1}$ to both sides of the high-order deformation equation (2.127), subject to the initial condition

$$u_m(\mathbf{x},0) = 0,$$

the m^{th} terms of the solution are respectively obtained in the following forms:

$$u_m^{DHAM1}(\mathbf{x},t) = \chi_m u_{m-1}^{DHAM1}(\mathbf{x},t) + \hbar exp(-\theta t)\int_0^t exp(\theta t)\mathcal{H}_1(\tau)R_m[\vec{u}_{m-1}^{DHAM1}]d\tau \tag{2.132}$$

and

$$u_m^{DHAM2}(\mathbf{x},t) = \chi_m u_{m-1}^{DHAM2}(\mathbf{x},t) + \frac{\hbar}{1+t}\int_0^t \mathcal{H}_2(\tau)R_m[\vec{u}_{m-1}^{DHAM2}]d\tau \tag{2.133}$$

where $R_m[\vec{u}_{m-1}]$ is as defined by (4.3). Indeed, the solution $u(\mathbf{x})$ of the original nonlinear equation (2.120), while using the linear operators $\mathcal{L}_1(u)$ and $\mathcal{L}_2(u)$ respectively, is expressed in the following form:

$$u^{DHAM1}(\mathbf{x},t) = \sum_{m=0}^{+\infty} a_m \exp(-mt) \tag{2.134}$$

and

$$u^{DHAM2}(\mathbf{x},t) = \sum_{m=0}^{+\infty} \frac{b_m}{(1+t)^m} \tag{2.135}$$

where the a_m, b_m's are coefficients depending on $\mathbf{x}$.

According to the rule of solution expression denoted by (2.134) and from equation (2.132), the auxiliary function $H(\tau)$ should be in the form $H(\tau) = e^{-k\tau}$, where k is an integer. It is found that, when $k \leqslant 1$, the solution of the high-order deformation

[6]The operator $\mathcal{L}_1(u)$ in equation (2.130) is a general case of the linear operator applied by Liao et. al.[99] where $\theta = 1$.

equation (2.132) contains the term te^{-t}, which incidentally disobeys the rule of solution expression (2.134). When $k > 2$, the base e^{-2t} always disappears in the solution expression of the high-order deformation equation (2.132), so that the coefficient of the term e^{-2t} cannot be modified even if the order of approximation tends to infinity. So, according to the so-called rule of coefficient ergodicity by Liao [99], we have to set $k = 2$, which uniquely determines the corresponding auxiliary function $\mathcal{H}_1(\tau) = exp(-2\tau)$.

Following the same discussion above for the rule of solution expression denoted by (2.135) and from equation (2.133), we get $\mathcal{H}_2(\tau) = \frac{1}{(1+\tau)^2}$. Thus, starting from the initial approximation $u_0(\mathbf{x}, t) = u_0(\mathbf{x})$, we can use the recurrence formulae (2.132) and (2.133) to successively obtain $u_m(\mathbf{x}, t)$ for $m \geqslant 1$ and at the same time the Mth order approximation has the form $u_M(\mathbf{x}, t) = \sum_{m=0}^{M} u_m(\mathbf{x}, t)$.

The HAM has been applied all these years for continuous functions because of the freedom to use different base functions to approximate a nonlinear problem, its validity for nonlinear problems and a convenient way to adjust the convergence region and the rate of the approximation series. The first attempt at a discrete version of HAM was made recently by Zhu et al. [183] for a diffusion equation. Unfortunately, applying it to a realistic nonlinear PDE models does not suit due to the analytical difficulties in working out the high order approximations demanded by HAM. Sorting this problem out whilst using the HAM's quality was the motivation of our work presented in Chapters 4. In this Chapter you can see how we can slightly modify and apply the method to image denoising and image segmentation tasks.

Chapter 3

Review of Variational Models for Image Restoration and Segmentation Techniques

3.1 Introduction

The calculus of variations deals with the theory of finding the maxima and minima of quantities defined as integrals containing unknown functions. The optimisation of appropriately chosen functionals with variational models involves the solution of nonlinear partial differential equations (PDEs) derived as necessary optimality conditions.

The history of calculus of variations begins with Newton, then initiated as a subject by the Bernoulli family. The first major contribution was made by the work of Euler, Lagrange and Laplace, with the classical problems of finding the path, curve, surface, etc., for which a given function has a stationary value. Further contributions were made in the nineteenth century by Hamilton, Dirichlet and Hilbert. In modern times, the calculus of variations has a wide-range of applications in classical solutions to minimization problems prescribed by boundary value problems. These problems involve certain types of differential equations, known as the associated Euler–Lagrange equations. Minimization problems that can be analyzed by the calculus of variations such as minimal curve length can be formulated as optimization problems. Such optimization problems are fundamental in many areas such as physics, engineering and all branches of mathematics.

In this chapter, we will introduce some basic mathematical analysis of nonlinear minimization principles and their application in image processing techniques. We explore the variational techniques, among the many existing approaches, due to the excellent results obtained from their applications. The main reason behind the success of variational PDE based models in image processing is due to the ease of imposing geometric regularity, such as smoothness. Various existing numerical methods for the realization of these models will be discussed. Mainly for two fundamental problems of image processing this thesis is concerned with: image denoising and digital image

segmentation. Both are important in real life due to their wide application in image restoration and reconstruction. From a mathematical point of view a large amount of research on these topics has been opened and many still remain unsolved.

Some mathematical analysis and properties of the total variation (TV) regularization functional are introduced. Some existing models used for solving the partial differential equation such as the Rudin-Osher-Fatemi (ROF) [135] model in denoising or Chan-Vese minimization arising in segmentation will be discussed briefly. The pros and cons of different smoothers (local and global) for solving PDEs arising from denoising and segmentation and other existing models are also described.

3.2 Denoising

Over the past decades, there has been extensive interest in denoising methods for noise removal in signals and images. The noise in digital images appears as random speckles on an otherwise smooth surface and significantly degrades image quality, making them look grainy or giving them a snowy appearance. Noise is caused from many possible sources such as sensitivity settings in the camera, temperature, light, length of exposure, fundamental limits of camera sets, etc. The magnitude of image noise can vary. The level of noise reaches higher levels for optical and radio astronomical images which are almost entirely noise and a small amount of information can be derived by sophisticated processing. Mathematically, the noise can be perceived as variations of random distribution without a particular pattern contaminating the brightness of an image. In many cases, this will reduce the quality of the image and is especially significant in the identification of small and low contrast objects.

There are many different types of noise and they may be classified as follows[1].

Amplifier Noise (Gaussian Noise)

This type of noise is additive noise, independent at each pixel and independent of the signal intensity. The values of noise have probability density function equal to that of the normal distribution, which is also known as the Gaussian distribution. It is caused primarily by thermal noise in the electronic components of digital cameras, such as the thermal vibrations of atoms in conductors, shot noise, black body radiation from the earth and other warm objects.

Uniform Noise

The uniform noise is caused by quantizing the pixels of a sensed image to a number of discrete levels. The noise is signal dependent andapproximately uniformly distributed.

[1] $http://en.wikipedia.org/wiki/Image\_noise$

Salt and Pepper Noise

Salt and pepper noise represents itself as randomly occurring dark pixels in bright regions and bright pixels in dark regions. This type of noise can mostly be caused by analog-to-digital converter errors, error concealment, dead pixels in the CCD digital camera, etc.

Shot Noise

Shot noise is a type of electronic noise which originates from the variation in the number of photons sensed at a given exposure level while current passes through PN junctions. Barrier crossings are random events, which bring random elementary current pulses which can be described as a Poisson distribution function, not very different from Gaussian.

Anisotropic Noise

This kind of noise is non-isotropic showing up with a significant orientation in images. For example, row or column noise is commonly found in image sensors and scratches in old films.

The noise can be additive [30] or multiplicative [45]. Speckle noise in the active radar and synthetic aperture radar (SAR) images is a multiplicative noise, a granular noise that inherently exists in and degrades the quality of these images. In this thesis we only consider images polluted with additive Gaussian noise. Removing the noise by using nonlinear PDE's as anisotropic diffusion filters has been shown to be more appropriate than traditional ways such as linear filters, which are simple to implement but may cause the restored image to be blurred at edges.

3.3 Variational Models and Partial Differential Equations

As mentioned in the introduction of this chapter we will deal with two image processing problems: digital image denoising and digital image segmentation. Although the digital images come from a continuous world they are defined in discrete settings. Discrete image processing methods can have easy techniques but continuous formulations of image processing methods are often more convenient to use and provide better results. Our interest in this theses centers on variational techniques.

We proceed now to explain the way variational techniques are formulated. Let us assume that a solution to a given image processing problem can be formulated as

$$u^* = argmin_{u \in S} F(u), \tag{3.1}$$

where u^* is an optimizer of the energy functional $F(\cdot)$, defined in an appropriate space S. If the functional F in equation (3.1) is continuous and differentiable, it is possible

to compute the first variation to determine the Euler-Lagrange equation

$$\frac{\partial F}{\partial u} = 0, \tag{3.2}$$

which gives a necessary condition for u^* to be an optimizer of F such that $\frac{\delta F}{\delta u}\Big|_{u^*} = 0$. The gradient descent (time marching) method computes an optimizer by introducing an artificial time t such that:

$$\frac{\partial u}{\partial t} = -\frac{\partial F}{\partial u}. \tag{3.3}$$

The denoising case can be an easy example to illustrate this. The goal in a denoising process is to recover an ideal image u from a noise observation u^0 (given available *data*). Let us assume that the degradation is due to an additive Gaussian noise $\eta = u^0 - u$ (unknown).

Assumption that the standard deviation σ of the noise is known or at least can be estimated we can the problem can be approached using variational techniques. One simple option to denoise the given image would be to minimize the L^2-norm of the noise, which would be

$$\min_u \left\{ \frac{1}{2} \int_\Omega |u^0 - u|^2 \, dxdy = \sigma^2 \right\}. \tag{3.4}$$

Equation (3.4) is an inverse problem and direct minimization will not guarantee a unique solution for u since many solutions (images) could satisfy the above equation. Therefore it becomes necessary to impose a *regularizer* for penalizing unwanted and irregular solutions using a priori knowledge. The *regularization* term is expected to improve the conditioning of the problem, bringing stability and possibly guaranteeing uniqueness of the solution. That is, we would like to solve a problem of the form

$$\min_u \quad F(u) \equiv \alpha \underbrace{\int_\Omega \Phi(u) \, dxdy}_{Regularization} + \underbrace{\frac{1}{2} \int_\Omega |u - u^0|^2 \, dxdy}_{Fitting}$$

One option for the regularizer would be the L^2-norm of the gradient of the given image, see G. Sapiro's book [141], and the variational model would be

$$\begin{aligned} u^* &= argmin_{u \in L^2(\mathbf{R}^2)} F(u) \\ F(u) &= \alpha \int_{\mathbf{R}^2} |\nabla u|^2 dx + \frac{1}{2} \int_\Omega |u - u^0|^2 \, dxdy, \end{aligned}$$

where F is in this case the Dirichlet functional. The Euler-Lagrange equation of this functional would be

$$\frac{\partial u^*}{\partial t} = \alpha \triangle u^* + |u^* - u^0|. \tag{3.5}$$

The solution of the partial differential equation (3.5) evolves in time t with an initial guess $u(0) = u^0$ and the process is stopped after a number of iterations, taking as the denoised image the current iterate u^*. For the segmentation case the variational model

is not very different and we will present two of these problems later in this chapter.

The selection of a regularizer plays a very important role since it imposes some properties to the resulting denoised image u. In the following we mainly consider the fundamental functionals such as Rudin-Osher-Fatemi (ROF) total variation [135] and the Mumford-Shah based functionals [115].

3.3.1 The Total Variation Model for Denoising

The success of PDE-based methods for denoising started with the anisotropic filters, so-called PDE-based filters and in particular the Perona-Malik model [126] which opened a completely new area of image processing. Since its appearance a lot of effort has been carried out to make use of anisotropic filters for image denoising and the requirement of preserving edges, a property that Perona-Malik model has been under attention. The model uses an edge stopping function $g(|\nabla u|^2) = \frac{1}{1+|\nabla u|^2}$ as diffusion coefficient evolving with the following equation:

$$\frac{\partial u}{\partial t} = \nabla u \cdot \Big(g(|\nabla u|^2) \nabla u \Big) \tag{3.6}$$

However the Perona-Malik model depends on the initial guess, regarded as ill-posed, and might have weak solutions. One should not expect uniqueness or stability for the model [127].

An edge preserving image denoising model with desirable mathematical properties was introduced by Rudin, Osher, and Fatemi (ROF) in their pioneering work [135]. The model is a total variation based image restoration designed with the explicit goal of preserving sharp discontinuities (edges) in images while removing noise and other unwanted fine scale detail. The ROF is the simplest convex variational model. The idea behind the model is to exhibit the reconstructed image as the minimizer of the following energy:

$$\min_u \left\{ \int_\Omega |\nabla u| \, dxdy + \frac{\lambda}{2} \int_\Omega |u - u^0|^2 \, dxdy \right\} \tag{3.7}$$

where λ is a positive parameter which selects the quantity of noise to be removed. The first term is the total variation of $u(x, y)$, a regularizer, while the second term is a fidelity term ensuring that the denoised image $u(x, y)$ will be close to the given image $u^0(x, y)$. The fitting parameter λ is important for balancing denoising and smoothing, therefore, it depends on the noise level. It can be observed that in the presence of high noise, the parameter λ has to be decreased. Large λ corresponds to very little noise removal and small λ yields a blurry, over smoothed restoration $u(x, y)$. Equation (3.7) is a well-posed problem so existence and uniqueness of its minimizer is guaranteed [29].

Minimization of (3.7) can be done by solving the nonlinear second order PDE

$$-\nabla \cdot \frac{\nabla u}{|\nabla u|} + \lambda(u - u^0) = 0 \ \text{ in } \ \Omega, \qquad (3.8)$$
$$\nabla u \cdot \boldsymbol{n} \ \text{on} \ \partial\Omega,$$

with $\boldsymbol{n}$ the unit outward normal. Although there exists very good solvers for the numerical solution of (3.8), see [168, 144, 143, 32] and references therein, there is still room for improvement.

An illustration of the TV model for denoising is presented in Figure 3.1 for a one dimensional signal case. From this figure we can notice a transformation of a piecewise smooth functions into piecewise constant functions, a phenomenon known as the staircase effect for the TV model.

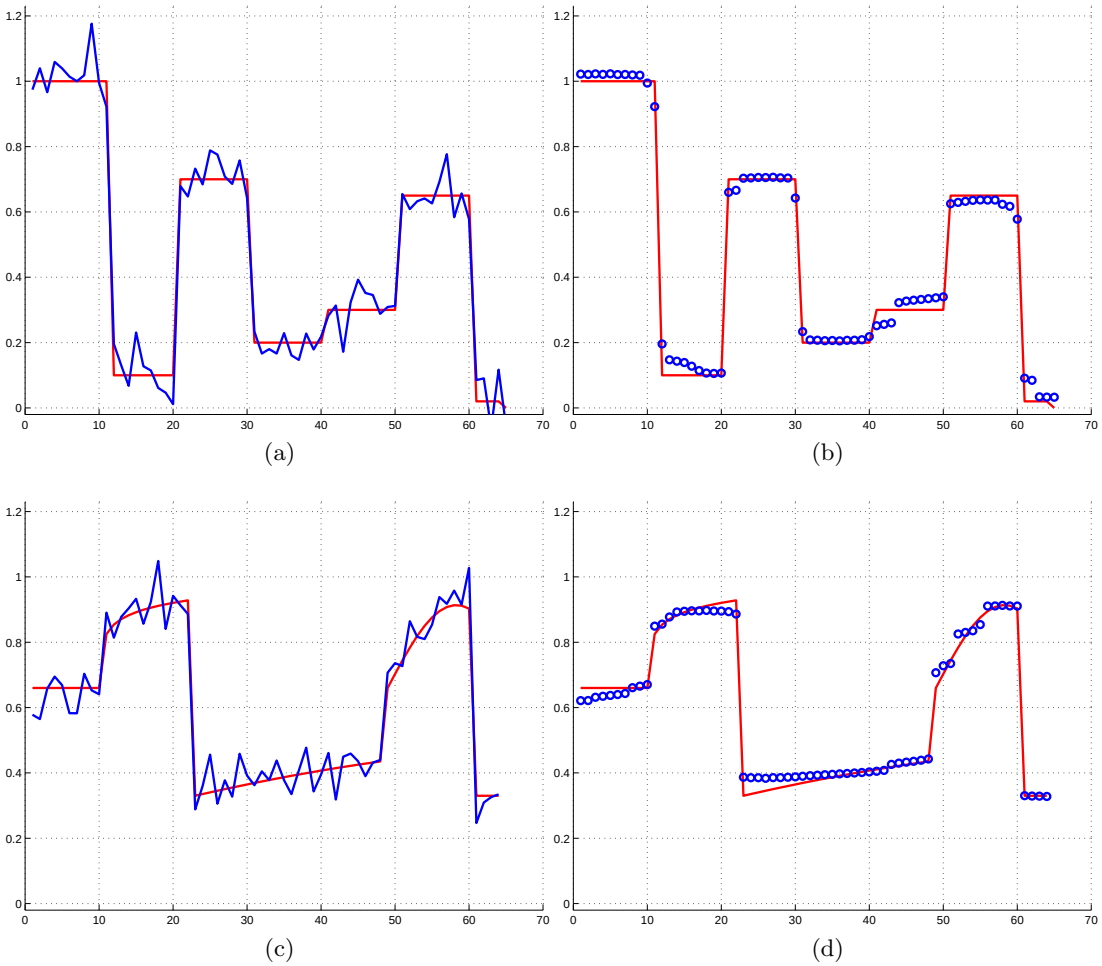


Figure 3.1: One dimensional signal restoration with TV model. Note that the original image in one spatial dimension is in red in all the images. (a) In blue is the noisy image in one spatial dimension for the given piece-wise constant, original signal in red. (b) In blue is a very good reconstruction with TV. (c) In blue the noisy image for a given original piecewise smooth image function. (d) TV recovered image in blue. The piecewise smooth function was transformed into a piece-wise constant function.

3.4 Image Segmentation

Image segmentation is defined as the process of partitioning a digital image into multiple patchworks of regions (sets of pixels) which have the same characteristics, such as intensity, texture, colour etc. [35, 9, 113, 152]. Typically the image segmentation is used to locate objects and boundaries (lines, curves, etc.) in a given image.

Identifying the regions in images that correspond to individual objects has broad applications in a variety of important fields such as computer vision, medical image processing, astronomy, aerospace, computer graphics, biometric security (fingerprints and other face identification), etc. Furthermore, many other applications appear every year in many different branches of technology and science.

During the past two decades, image segmentation applications have attracted the attention of many mathematicians, and a considerable number of techniques have been developed since then. There are basically two main approaches for image segmentation: the computational (non-variational) approach and the variational or energy-based approach. A simple way to partition an image can be done with the non-variational based methods such as threshold techniques [107, 146, 167], region merging algorithms [3, 188], the watershed segmentation techniques [52, 19] and so on. Most of them are based on a discrete setting, which makes them dependent on parametrization, and are not defined in a rigorous mathematical framework, see [153]. Among these techniques found in the literature, the variational PDE approach has been found to be very successful as a great way of formulating the segmentation model and representing the geometry of the segmented partitions. In what follows, we detail some techniques and in particular the level set variational model related to Mumford-Shah image segmentation [115].

3.4.1 Variational Image Segmentation Models

Given an observed image u_0 in a bounded domain Ω we are looking for a meaningful edge set Γ which leads to a complete partition of Ω into regions $\{\Omega_k, k = 1, 2, \ldots\}$ in $\Omega \backslash \Gamma$. The partitions Ω_k represent individual "objects", corresponding to at one real physical object or pattern in our 3-D world. In this way the given image u_0 can be represented as $[u_k, \Omega_k]$ with u_k as the region's intensity values.

In recent years, a class of image segmentation models based on variational approaches have been introduced and found to be successful due to their robustness and reliability. Those models are defined in a continuous setting and mathematically well studied. These models are formulated as energy minimization problems and can be categorized broadly into edge-based models [28, 83, 84, 92] and region-based models [115, 37, 39, 132, 166]. Edge-based models use the edge information guiding the active contours towards the object boundary and the region based ones make use of image intensities to guide the motion of active contours. We shall only review a few models below that are directly related to our work.

In the same way as denoising, the structure of the variational models for solving

image segmentation problems can be formulated as

$$u^* = argmin_{u \in S} F(u), \tag{3.9}$$

where u is an optimizer and the functional $F(\cdot)$ gives the solution to the given image segmentation problem.

The early well known variational image segmentation models are the Mumford-Shah model [115] and the active contour model of Kass-Witkin-Terzzopoulos [83]. In what follows, we detail some technique related image segmentation models.

3.4.2 Mumford-Shah Approach

The seminal work of Mumford and Shah [115] has been followed by an impressive number of variational formulations of theoretical, methodological, and practical studies [37, 140, 166, 124, 10, 85, 113, 41, 110, 156, 110, 18].

Let Ω be a bounded domain in $\mathbb{R}^n$ and u_0 be a bounded measurable function defined on Ω (representing the given image). The Mumford and Shah functional concerns pairs (u, Γ) where Γ is a closed subset and u is a function belonging to $\mathcal{C}^1(\Omega \setminus \Gamma)$. The n-dimension Mumford-Shah functional can be defined as

$$F^{MS}(u, \Gamma) = \alpha \mathcal{H}^{n-1}(\Gamma) + \beta \int_{\Omega} (u - u_0)^2 dxdy + \int_{\Omega \setminus \Gamma} |\nabla u|^2 dxdy, \tag{3.10}$$

where $\mathcal{H}^{n-1}$ is the $(n-1)$-dimensional Hausdorff measure[2], α and β are positive (tuning) parameters. Let (u, Γ) be a minimizing pair of $F^{MS}(u, \Gamma)$. The first terms is the constraint on the set of discontinuities (edges) Γ, which requires the boundaries to be as short as possible (and therefore as smooth as possible). The second term is a fitting term with respect to the given image u_0, forcing u to approximate u_0. The last term is the regularization term which asks u not to vary much on $\Omega \setminus \Gamma$. Theoretical results on the existence and regularity of minimizers of (3.10) can be found in [115, 112, 113].

The model aims to draw smoothly the objects (u, Γ), which essentially is an idealization of u_0. Created simply the image can be sorted as a cartoon of the original image u_0. In other words assume that u can be considered as a piecewise constant function, i.e $u = c_k$ inside each connected region Ω_k. The Mumford and Shah model reduces to the following functional:

$$F_{RC}^{MS}(u, \Gamma) = \alpha \oint_{\Gamma} ds + \beta \sum_k \int_{\Omega_k} (u_0 - c_k)^2 dxdy. \tag{3.11}$$

where the first term of (3.11) is the length of the set Γ, and an option for c_k is the mean value of each region Ω_k. In practice, because of the unknown set Γ, the functionals (3.10) and (3.11) are difficult to minimize, are of lower dimension and are not convex.

[2]The usual $(n-1)$-dimensional Hausdorff measure is $(n-1)$-dimensional area in case of subsets of regular hyper surfaces. In the most relevant case $n = 2$ it is the length.

A possible solution to these problems was introduced by Chan-Vese [37] and will be addressed in Subsection §3.4.6.

3.4.3 Snake: Active Contour Model

One of the first PDE based models, the so-called classical active contour/snake model, was initially proposed by Kass,Witkin and Terzopolous [83].

The model minimizes a curve-based objective functional towards a regular curve which locates sharp image intensity variations (edges). Let Ω be a bounded and open set in $\mathbb{R}^n$, with $\partial\Omega$ its boundary. Let u_0 be a given image, as a bounded function defined on $\overline{\Omega}$ and with real values. Denote by $C(s) : [0,1] \to \mathbb{R}^n$ a piecewise $\mathcal{C}^1([0,1])$ planar curve, represented as $C(s) = (x(s), y(s)) \in \Omega$, $s \in [0,1]$. The snakes model is to minimize the following functional:

$$F_{KWT}(C(s)) = \alpha \int_0^1 |\frac{\partial C(s)}{\partial s}|^2 ds + \beta \int_0^1 |\frac{\partial^2 C(s)}{\partial^2 s}|^2 ds + \lambda \int_0^1 g(\nabla u_0(C(s)))^2 ds, \quad (3.12)$$

where α, β and λ are positive constants. The first two term in (3.12) are the internal energy and control the elastic and bending energies, characteristic of the contour itself. The minimization of these energy terms serve to ensure that the curve remains smooth and continuous. The third term is the external energy and attracts the contours towards the edge of the object in the image u_0. The function $g(|\nabla u_0|)$ is called an edge detector. For a given image u_0, its gradient image ∇u_0 has high values in the neighborhood of the object image, due to the intensity change. Clearly $g(|\nabla u_0|)$ is strictly positive in homogenous regions and near zeros on the vicinity of an edge. As a result the energy minimization procedure will push the contour towards the edges, even if the contour is initialized far away from the object boundary. For example $g : [0, +\infty[\to \mathbb{R}^+$ is given by

$$g(\nabla u_0) = \frac{1}{1 + \gamma |\nabla (u_0 * G_\sigma)|^2}, \quad (3.13)$$

where γ is a positive constant and $u_0 * G_\sigma$ is a smoother version of u_0, obtained as a convolution of the image u_0 with the Gaussian function with standard deviation σ and mean μ, $G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp^{-|(x-\mu)^2+(y-\mu)^2|/2\sigma^2}$. The functional $F_{KWT}(C(s))$ is not convex [83], so the solution will not be unique and it has local minima. The local minima of F_{KWT} can be reached by solving the following Euler-Lagrange equation

$$-\alpha \frac{\partial^2 C}{\partial s^2} + \beta \frac{\partial^4 C}{\partial s^4} + \lambda \nabla g^2 = 0. \quad (3.14)$$

A modification of the snakes model was proposed by Cohen [46] adding a pressure force that helps it to avoid local minima. Later Caselles, Kimmel and Sapiro [28] proposed a geodesic active contour model, which we will introduce in Subsection §3.4.5. Although these models are quite powerful, they are strongly dependent on the initialization step, the parametrization of the curve C and need special procedures to handle topological

changes on the evolving curve.

A powerful tool which deals with the topology change is the level set method [122, 121, 150] introduced in Subsection §2.6. As we will show in the following the curve C is implicitly represented by a level set function of higher dimension ϕ, and the curve evolution equation can be re-written in a level set formulation.

3.4.4 Geometric Characteristics and Contour Representation with Level Sets

For a given domain $\Omega \subset \mathbb{R}^n$, with *interface* $\Gamma = \partial\Omega$ the domain is n-dimensional, while the interface has dimension $n-1$. The main idea in the level set method is to implicitly represent an interface Γ (let's say for example in $\mathbb{R}^2$) as a level set of a function ϕ (level set function) of higher dimension (in this case $\mathbb{R}^3$) and compute the geometric characteristics and the motion of the front with this level set function. We recall from Section §2.6 that the level set function ϕ of the closed front Γ is defined as follows [121]:

$$\begin{cases} \phi(x,y,t) > 0 \text{ inside } \Gamma, \\ \phi(x,y,t) < 0 \text{ outside } \Gamma, \\ \phi(x,y,t) = 0 \text{ on } \Gamma. \end{cases}$$

At time $t = 0$, the curve is the zero level set where $\Gamma_0 = \Gamma(p, t = 0) = \{(x, y) : \phi(x, y, 0) = 0\}$, and at later times the curve is the new zero level set of a function (x, y, t), $\Gamma(p, t) = \{(x, y) : \phi(x, y, t) = 0\}$ (or $\Gamma(t) = \phi^{-1}(0)$).

For simplicity we do not consider the parametric/explicit representation of a contour $C(s)$ but its geometry/implicit representation. In this way the area of the region inside Γ and the length of the interface Γ are

$$\begin{cases} Area(\Omega) = \int_{\mathbb{R}^2} H(\phi)dxdy \\ Lenghth(\Gamma) = \int_{\mathbb{R}^2} |\nabla H(\phi)|dxdy = \int_{\mathbb{R}^2} \delta(\phi)|\nabla\phi|dxdy, \end{cases}$$

where δ and H are the Dirac delta and the Heaviside functions. The same equalities can be found for a $\Omega \subset \mathbb{R}^3$

$$\begin{cases} Volume(\Omega) = \int_{\mathbb{R}^3} H(\phi)dxdydz \\ Area(\Gamma) = \int_{\mathbb{R}^3} |\nabla H(\phi)|dxdydz = \int_{\mathbb{R}^3} \delta(\phi)|\nabla\phi|dxdydz. \end{cases}$$

The above equalities are basic for all the level set methods which are going to not only be in the following of this chapter but for the rest of the thesis.

3.4.5 The Geodesic Active Contours Model

Based on Kass et al [83], Caselles, Kimmel and Sapiro [28] proposed a geodesic active contour model:

$$F_{GAC}(C(s)) = \int_0^1 g(|\nabla u_0(C(s))|)|C'(s)|ds, \tag{3.15}$$

that is invariant with respect to a new curve parametrization. Replacing the Euclidean length of the curve C, $L(C) = \int_0^1 |C'(s)|ds = \int_0^{L(C)} ds$, where ds is the Euclidean element of length, equation (3.15) becomes

$$F_{GAC}(C(s)) = \int_0^{L(C)} g(|\nabla u_0(C(s))|)ds. \tag{3.16}$$

The function g is the edge detecting function defined in the same way as (3.13). The functional (3.16) has Euclidean length ds weighted by the function g which contains information concerning the boundaries of objects [10]. The equivalence between F_{GAC} and F_{KWT} at $\beta = 0$ was studied in [28]. The functional F_{GAC} decreases most rapidly in the following direction [28]:

$$\frac{\partial C}{\partial t} = g\kappa\overrightarrow{n} - (\nabla g \cdot \overrightarrow{n})\overrightarrow{n}, \tag{3.17}$$

where κ is the Euclidean curvature and $\overrightarrow{n}$ is the unit normal vector. The steady state solution of (3.17) in terms of the variational level set formulation is

$$\frac{\partial \phi}{\partial t} = |\nabla \phi|(\nabla \cdot (g\frac{\nabla \phi}{|\nabla \phi|}) + \nu g), \tag{3.18}$$

where ϕ is a Lipschitz function representing C as a zero level set. Here the constant ν is added to increase the evolution speed and attract the curves towards the boundary and contributes an extra area-based speed.

The snakes/active contour models depend on the gradient of the image. In practice the discrete gradients are bounded and then the stopping function g is never zero at the edges, and the contour may pass through the image edges [37]. On the other hand if the given image is too noisy, then the isotropic smoothing Gaussian has to be strong. This will cause smooth of the edges as well.

In the next section we describe another active contour model which does not depend on the edge function to stop the contour at edges, i.e. a model which is not based on the gradient of the image for the stopping process. This is the Chan-Vese model [37], "Active Contour without Edges".

3.4.6 Chan-Vese Model

The Chan-Vese variational model [39] of active contours without edges has been used successfully for segmentation of all image objects/features. This model does not use the gradient of the image as a stopping process as it is a region based method and the stopping term is based on the Mumford and Shah segmentation technique [115]. For more other variational models, refer to [36, 92, 111, 102, 149] among others.

The basic idea of the model is as follows. Assume that a given image u_0, a bounded function, is formed by two regions of approximatively piecewise constant intensities, of distinct values u_0^i and u_0^o. The object/objects to be detected are represented by the

region with intensities closest to the value u_0^i. Let Γ denote the boundary that separates the two regions Ω_1 and $\Omega_2 = \Omega \backslash \overline{\Omega_1}$. Then $u_0 \approx u_0^i$ inside the object (inside Γ) and $u_0 \approx u_0^o$ outside the object (outside Γ). Chan and Vese [39] proposed the variational problem

$$\inf_{c_1, c_2, \Gamma} F^{CV}(\Gamma, c_1, c_2) \tag{3.19}$$

for the segmentation of all image objects/features, where

$$\begin{aligned} F^{CV}(\Gamma, c_1, c_2) = \mu(length(\Gamma))^n + \nu area(inside(\Gamma)) + \\ \lambda_1 \int_{inside(\Gamma)} |u_0 - c_1|^2 dxdy + \lambda_2 \int_{outside(\Gamma)} |u_0 - c_2|^2 dxdy. \end{aligned} \tag{3.20}$$

where c_1 and c_2 are the average values of u_0 inside and outside the variable contour Γ, also μ, ν, λ_1 and λ_2 are non-negative fixed parameters that should be related to the contour features. The "$length(\Gamma)^n$" means the Hausdorff $(n-1)$-dimensional measure $\mathcal{H}^{n-1}(\Gamma)$. If the curve is forced to move only inside or outside then we take ν positive, otherwise $\nu = 0$. In the original paper of Chan-Vese [37] $\lambda_1 = \lambda_2 = 1$ and $n = 1$, which will be considered for the rest of this chapter. Improvement of Chan-Vese [37] has been continuously made [161, 39, 76, 33, 166] etc.

This model is a special case of Mumford and Shah segmentation model [115], restricted to only two regions with constant average intensities. The initial curve can be placed somewhere in the image, and it detects contours both with or without gradient, with very smooth boundaries or even with discontinuous boundaries. A multiphase formulation of the Chan-Vese model can be found in [166] and an extension of the two phase piecewise constant CV model to piecewise linear and smooth segmentation in [164].

We have to acknowledge that because the energy is nonconvex (allowing therefore many local minima) the solution may depend on the initial curve in some cases. To avoid this drawback new variational models and techniques have been proposed, from which we will overview two; see Subsections §3.4.11 and 3.4.9.

Chan-Vese [37] has a level set formulation, which means there is potential for numerical application of the model in practice.

3.4.7 Level Set Formulation for the Chan-Vese Model:

As both the integral and the limits of integration in equation (3.20) are not known, to overcome this problem, a level set function is introduced. For simplicity lets consider that the image u_0 is defined in a 2-dimensional domain, i.e. $\Omega \subset \mathbb{R}^2$. The unknown curve Γ can be represented by the zero level set of a Lipschitz function [122] $\phi : \mathbb{R}^2 \to \mathbb{R}$

such that

$$\begin{cases} \Gamma = \partial\Omega_1 = \{(x,y) \in \Omega \;\Big|\; \phi(x,y) = 0\}, \\ \text{inside}(\Gamma) = \Omega_1 = \{(x,y) \in \Omega \;\Big|\; \phi(x,y) > 0\}, \\ \text{outside}(\Gamma) = \Omega_2 = \{(x,y) \in \Omega \;\Big|\; \phi(x,y) < 0\}. \end{cases}$$

In this way the unknown lower dimensional variable curve Γ is replaced by another unknown higher dimensional variable ϕ. A typical example of a level set function is given by the signed distance function to the curve. Moreover let us define the Heaviside and the Dirac delta functions as

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \qquad \text{and} \qquad \delta(x) = H'(x).$$

Each term of the energy $F^{CV}(\Gamma, c_1, c_2)$ can be easily expressed in terms of ϕ :

$$\begin{aligned} length\{\phi = 0\} &= \int_\Omega |\nabla H(\phi)| dxdy = \int_\Omega \delta(\phi)|\nabla\phi| dxdy, \\ area\{\phi \geq 0\} &= \int_\Omega H(\phi) dxdy, \\ \int_{\phi \geq 0} |u_0 - c_1|^2 dxdy &= \int_\Omega |u_0 - c_1|^2 H(\phi) dxdy, \\ \int_{\phi < 0} |u_0 - c_2|^2 dxdy &= \int_\Omega |u_0 - c_2|^2 (1 - H(\phi)) dxdy. \end{aligned}$$

In the level set formulation equation (3.20) is rewritten in the following way

$$\begin{aligned} F^{CV}(\phi, c_1, c_2) = \mu \int_\Omega \delta(\phi)|\nabla\phi| dxdy + \nu \int_\Omega H(\phi) dxdy + \\ \lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H(\phi(x,y)) dxd + \\ \lambda_2 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H(\phi(x,y))) dxdy. \end{aligned} \tag{3.21}$$

Once the level set function ϕ is obtained, the segmented image is given by

$$u = c_1 H(\phi) + c_2(1 - H(\phi)).$$

Details for the existence of minimizers are discussed in Chan-Vese paper [37].

The minimization of (3.21) can be easily obtained by decoupling the variables. Minimize $F^{CV}(\phi, c_1, c_2)$ with respect to c_1, c_2, keeping $\phi(x,y)$ fixed, we have

$$c_1(\phi(x,y)) = \frac{\int_\Omega u_0(x,y) H(\phi(x,y)) dxdy}{\int_\Omega H(\phi(x,y)) dxdy}, \tag{3.22}$$

if $\int_\Omega H(\phi(x,y)) dxdy > 0$ (i.e the curve has a nonempty interior in Ω), and

$$c_2(\phi(x,y)) = \frac{\int_\Omega u_0(x,y)(1 - H(\phi(x,y))) dxdy}{\int_\Omega (1 - H(\phi(x,y))) dxdy}, \tag{3.23}$$

if $\int_\Omega (1 - H(\phi(x,y)))dx > 0$ (i.e the the curve has a nonempty exterior in Ω).

To compute the Euler-Lagrange equation for the unknown function ϕ, as H is not differentiable at 0, we consider regularized versions of H and δ functions, denoted by H_ϵ and δ_ϵ respectively [37, 42, 20]:

$$H_\epsilon(x) = \frac{1}{2}(1 + \frac{2}{\pi}\arctan(\frac{x}{\epsilon})), \qquad \delta_\epsilon(x) = H'_\epsilon(x) = \frac{\epsilon}{\pi(\epsilon^2 + x^2)}, \tag{3.24}$$

where $H_\epsilon \to H$ when $\epsilon \to 0$. The regularized functional of $F^{CV}(\phi, c_1, c_2)$ is given by

$$\begin{aligned} F^{CV}_\epsilon(\phi, c_1, c_2) = \mu \int_\Omega \delta_\epsilon(\phi(x,y))|\nabla\phi|dxdy + \nu \int_\Omega H_\epsilon(\phi)dxdy + \\ \lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H_\epsilon(\phi(x,y))dxdy + \\ \lambda_2 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H_\epsilon(\phi(x,y)))dxdy. \end{aligned} \tag{3.25}$$

and the minimization problem is

$$\inf_{\phi, c_1, c_2} F^{CV}_\epsilon(\phi, c_1, c_2).$$

Let us keep c_1 and c_2 fixed, and minimize F^{CV}_ϵ with respect to the unknown ϕ. Using the Gâteaux derivatives to find the first variation of the functional F^{CV}_ϵ with respect to ϕ let us choose ψ as a test function of the same type as ϕ and find:

$$\lim_{h \to 0} \frac{dF^{CV}_\epsilon(\phi + h\psi, c_1, c_2)}{dh} = 0.$$

Using the notation ϕ, u_0 instead of $\phi(x,y)$, $u_0(x,y)$ we have

$$\begin{aligned} &\frac{d}{dh}\int_\Omega \mu\Big(\delta_\epsilon(\phi + h\psi)|\nabla(\phi + h\psi)|dxdy\Big|_{h=0} + \frac{d}{dh}\nu\int_\Omega H_\epsilon(\phi + h\psi)dxdy\Big|_{h=0+} \\ &\frac{d}{dh}\int_\Omega \Big(\lambda_1|u_0 - c_1|^2 H_\epsilon(\phi + h\psi) + \lambda_2|u_0 - c_1|^2(1 - H_\epsilon(\phi + h\psi))\Big|_{h=0} = 0 \end{aligned}$$

$$\begin{aligned} &\Rightarrow \mu\int_\Omega \Big(|\nabla(\phi + h\psi)|\frac{d}{dh}\delta_\epsilon(\phi + h\psi)dxdy + \delta_\epsilon(\phi + h\psi)\frac{d}{dh}|\nabla(\phi + h\psi)|\Big)dxdy\Big|_{h=0} + \\ &\int_\Omega \Big(\nu\delta_\varepsilon(\phi + h\psi)\psi\Big)dxdy\Big|_{h=0} + \int_\Omega \Big(\lambda_1(u_0(x,y) - c_1)^2\delta_\varepsilon(\phi + h\psi)\psi - \\ &\lambda_2(u_0(x,y) - c_1)^2\delta_\varepsilon(\phi + h\psi)\psi\Big)dxdy\Big|_{h=0} = 0. \end{aligned}$$

Computing the derivatives we have:

$$\mu \int_\Omega \left(\delta'_\epsilon(\phi)|\nabla\phi|\psi + \delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi \right) dxdy + \\ \int_\Omega \delta_\epsilon(\phi)\Big(\nu + \lambda_1(u_0(x,y)-c_1)^2 - \lambda_2(u_0(x,y)-c_1)^2\Big)\psi dxdy = 0$$

or

$$\mu \int_\Omega \delta'_\epsilon(\phi)|\nabla\phi|\psi dxdy + \mu\int_\Omega \delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi dxdy + \\ \int_\Omega \delta_\epsilon(\phi)\Big(\nu + \lambda_1(u_0(x,y)-c_1)^2 - \lambda_2(u_0(x,y)-c_1)^2\Big)\psi dxdy = 0 \tag{3.26}$$

From Green's theorem we have

$$\int_\Omega v\nabla\cdot\vec{w}dx = -\int_\Omega \nabla v\cdot\vec{w}dx + \int_{\partial\Omega} v\vec{w}\cdot\vec{n}ds.$$

Considering this theorem for the second integral in (3.26) with

$$\psi = v \qquad and \qquad \frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi = \vec{w},$$

we can rewrite it as follows

$$\int_\Omega \psi\nabla\cdot\Big(\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi\Big)dxdy = -\int_\Omega \nabla\psi\cdot\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi dxdy + \int_{\partial\Omega}\psi\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi\cdot\vec{n}ds,$$

which implies that

$$\int_\Omega \delta_\epsilon(\phi)\frac{\nabla\phi\cdot\nabla\psi}{|\nabla\phi|}dxdy = -\int_\Omega \psi\nabla\cdot\Big(\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi\Big)dxdy + \int_{\partial\Omega}\psi\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}ds,$$

where $\nabla\phi\cdot\vec{n} = \dfrac{\partial\phi}{\partial n}$. Thus (3.26) becomes

$$\mu\int_\Omega \delta'_\epsilon(\phi)|\nabla\phi|\psi - \mu\int_\Omega \psi\nabla\cdot\Big(\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi\Big)dxdy + \mu\int_{\partial\Omega}\psi\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}ds + \\ \int_\Omega \delta_\epsilon(\phi)(\nu + \lambda_1(u_0-c_1)^2 - \lambda_2(u_0-c_2)^2)\psi dxdy = 0,$$

implying that

$$\mu\int_\Omega \delta'_\epsilon(\phi)|\nabla\phi|\psi dxdy - \mu\int_\Omega \delta_\epsilon(\phi)\nabla\cdot\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)\psi dxdy - \mu\int_\Omega \delta'_\epsilon(\phi)\nabla\phi.\frac{\nabla\phi}{|\nabla\phi|}\psi dxdy + \\ \mu\int_{\partial\Omega}\psi\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}ds + \int_\Omega \delta_\epsilon(\phi)(\nu + \lambda_1(u_0-c_1)^2 - \lambda_2(u_0-c_2)^2)\psi dxdy = 0.$$

Finally we obtain

$$-\int_{\Omega}\mu\delta_{\epsilon}(\phi)\nabla\cdot\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)\psi dxdy+\int_{\partial\Omega}\mu\frac{\delta_{\epsilon}(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}\psi ds+$$
$$\int_{\Omega}\delta_{\epsilon}(\phi)(\nu+\lambda_1(u_0-c_1)^2-\lambda_2(u_0-c_2)^2)\psi dxdy=0,$$

for all test functions ψ. Choosing $\psi\in C_c^1(\Omega)$ which is arbitrary, we deduce the following Euler-Lagrange equation for ϕ:

$$\begin{cases}\delta_{\epsilon}(\phi)\Big[\mu\nabla\cdot\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)-\nu-\lambda_1(u_0-c_1)^2+\lambda_2(u_0-c_2)^2\Big]=0 & \text{in }\Omega,\\ \\ \dfrac{\delta_{\epsilon}(\phi)}{|\nabla\phi|}\dfrac{\partial\phi}{\partial n}=0 \qquad \text{or} \qquad \dfrac{\partial\phi}{\partial n}=0 & \text{on }\partial\Omega.\end{cases}\tag{3.27}$$

The approximation of equation (3.27) can be done by introducing an artificial time step t and using the gradient descent method, as considered by the authors [37]. In this way we get the following evolution equation:

$$\begin{cases}\frac{\partial\phi}{\partial t}=\delta_{\epsilon}(\phi)\Big[\mu\nabla.\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)-\nu-\lambda_1(u_0-c_1)^2+\lambda_2(u_0-c_2)^2\Big] & \text{in }\Omega,\\ \phi(t,x,y)=\phi_0(x,y) & \text{in }\Omega,\\ \dfrac{\partial\phi}{\partial n}=0 & \text{on }\partial\Omega.\end{cases}\tag{3.28}$$

A standard re-scaling can be made, as in Osher and Sethian [122] and Zhao et al. [182], by replacing $\delta_{\epsilon}(\phi)$ with $|\nabla\phi|$. This re-scaling does not affect the steady state solution, but it does remove the stiffness of the zero level sets of ϕ. In the following we will use $\delta_{\epsilon}(\phi)$ as in [37]. In the next section we derive the numerical approximation of the model and will discuss the existing methods used for solving and improving the above evolution problem (3.28).

3.4.8 Numerical Methods for the Chan-Vese Model

We shall follow the same procedure of replacing the non-differentiable H function with a regularized Heaviside function H_{ϵ}, as in [182, 10, 37]. Different regularized Heaviside functions can be used e.g.:

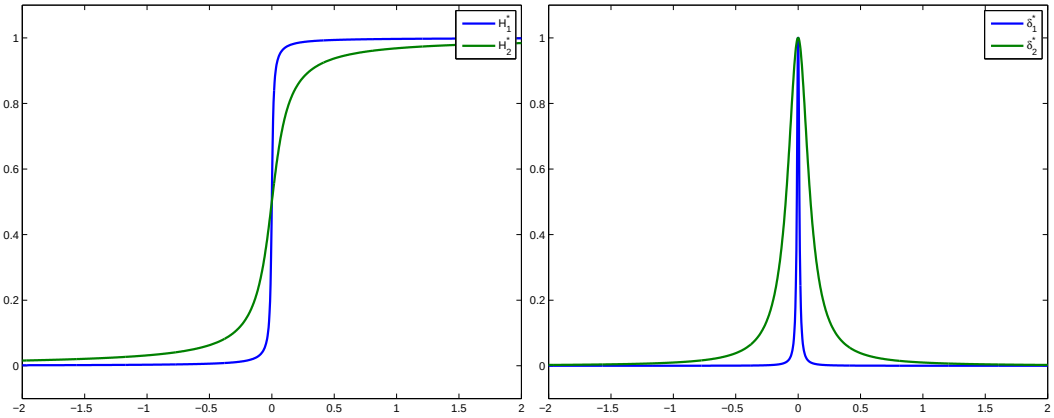
$$H_{1\epsilon}(z)=\begin{cases}0, & z<-\epsilon\\ \frac{1}{2}\left[1+\frac{z}{\epsilon}+\frac{1}{\pi}\sin(\frac{\pi z}{\epsilon})\right], & |z|\leq\epsilon,\\ 1, & z>\epsilon,\end{cases}\qquad \delta_{1\epsilon}(z)=\begin{cases}0, & z<-\epsilon\\ \frac{1}{2}\left[\frac{1}{\epsilon}+\frac{1}{\epsilon}\cos(\frac{\pi z}{\epsilon})\right], & |z|\leq\epsilon,\\ 1, & z>\epsilon,\end{cases}$$

$$H_{2\epsilon}(z)=\frac{1}{2}(1+\operatorname{erf}(\frac{\epsilon}{z})),\qquad \delta_{2\epsilon}(z)=(\frac{1}{\epsilon}\pi^{1/2}\exp(-\frac{x^2}{\epsilon^2}),$$

$$H_{3\epsilon}(z)=\frac{1}{1+e^{-\frac{2z}{\epsilon}}},\qquad \delta_{3\epsilon}(z)=-2\frac{e^{-\frac{2z}{\epsilon}}}{\epsilon(1+e^{-\frac{2z}{\epsilon}})^2},$$

$$H_{4\epsilon}(z) = \frac{1}{2}(1 + \frac{2}{\pi}\arctan(\frac{z}{\epsilon})), \qquad \delta_{4\epsilon}(z) = \frac{1}{\epsilon\pi(1 + x^2/\epsilon^2)},$$

where error function $\text{erf}(x)$ is twice the integral of the Gaussian distribution with 0 mean and variance of $\frac{1}{2}$ in the form $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$. Figure 3.2 shows a graphical comparison between two different regularized Heaviside functions (an approximation to H) and the derivative approximation δ_ϵ of the Delta function. For a given ϵ we can notice



(a) H_ϵ^* approximation to the Heaviside function H (b) δ_ϵ^* approximation to the Delta function δ

Figure 3.2: Approximation of the Heaviside and Delta function with H_ϵ and δ_ϵ.

different smoothing support from the corresponding H_ϵ approximation and this changes within the function by increasing or decreasing the ϵ. For the given $H_{1\epsilon}, H_{2\epsilon}, H_{3\epsilon}$ and $H_{4\epsilon}$ above and their corresponding Delta functions $\delta_{1\epsilon}, \delta_{2\epsilon}, \delta_{3\epsilon}$ and $\delta_{4\epsilon}$ the differences between them are that the first three functions have a small support in the interval $[-\epsilon, \epsilon]$, while $H_{4\epsilon}$ and its corresponding $\delta_{4\epsilon}$ are different from zero everywhere. Since the Chan-Vese energy is nonconvex (allowing therefore many local minima), the solution may depend on the initial curve. Using $H_{1\epsilon}$ and $\delta_{1\epsilon}$ for the Chan-Vese algorithm the Euler–Lagrange equation for ϕ acts locally on a few level curves around $\{\phi = 0\}$, so the local minimizer of the energy can sometimes be computed. To avoid this, since the Chan-Vese algorithm has the tendency to compute a global minimizer, using $H_{4\epsilon}$ and $\delta_{4\epsilon}$ which are different from zero everywhere, the Euler–Lagrange equation will act on all level curves and avoid the local calculation of the energy. In this way, in practice, we can obtain a global minimizer, independently of the position of the initial curve. Adjustment of ϵ might be needed for the selective segmentation as we will see in Chapter 5 since a local minimum is required.

In the following we will show the discretization of the Chan-Vese equation in ϕ,

using finite differences in an implicit scheme for

$$\begin{cases} \phi(0,x,y) = \phi_0(x,y) & \text{in } \Omega, \\ c_1(\phi) = \dfrac{\int_\Omega u_0 H_\epsilon(\phi) dxdy}{\int_\Omega H_\epsilon(\phi) dxdy} \qquad c_2(\phi) = \dfrac{\int_\Omega u_0(1-H_\epsilon(\phi)) dxdy}{\int_\Omega (1-H_\epsilon(\phi)) dxdy} & \\ \dfrac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\Big[\mu \nabla.\Big(\dfrac{\nabla \phi}{|\nabla \phi|}\Big) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\Big] & \text{in } \Omega, \\ \dfrac{\delta_\epsilon(\phi)}{|\nabla \phi|}\dfrac{\partial \phi}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{3.29}$$

After $c_1(\phi)$ and $c_2(\phi)$ have been computed from equation (3.29) we fix them and then solve the PDE for ϕ. Once ϕ is found, then update $c_1(\phi)$ and $c_2(\phi)$ and so on. To solve the above PDE in equation (3.29), we first recall the usual notations: let the size of given image u_0 be $m_1 \times m_2$, and let h_1 and h_2 be the space step in x and y direction, $\triangle t$ the time step and $(x_i, y_j) = (ih_1, jh_2)$, for $i = 1, \ldots, m_1$ and $j = 1, \ldots, m_2$ the grid points. Let $\phi^k_{i,j} = \phi(k\triangle t, x_i, y_j)$ be an approximation of $\phi(t,x,y)$, where $k \geq 0$ and $\phi^0 = \phi_0$ will be given (initial guess). The finite differences are denoted by

$$\begin{aligned} \triangle^x_- \phi_{ij} &= \phi_{ij} - \phi_{i-1,j}, \ \triangle^x_+ \phi_{ij} = \phi_{i+1,j} - \phi_{ij}, \\ \triangle^y_- \phi_{ij} &= \phi_{ij} - \phi_{i,j-1}, \ \triangle^y_+ \phi_{ij} = \phi_{i,j+1} - \phi_{ij}. \end{aligned} \tag{3.30}$$

For a given ϕ^k, first compute $c_1(\phi^k)$ and $c_2(\phi^k)$ and then discretize the above parabolic PDE (3.29), compute ϕ^{k+1} by the following discretization and linearization of (3.29) in ϕ

$$\begin{aligned} \frac{\phi^{k+1}_{ij} - \phi^k_{ij}}{\triangle t} = \delta_\epsilon(\phi^k_{ij})\Big[& \frac{\mu}{h_1^2}\triangle^x_-\Big(\frac{\triangle^x_+ \phi^{k+1}_{ij}}{\sqrt{(\triangle^x_+ \phi^k_{ij}/h_1)^2 + ((\phi^k_{i,j+1} - \phi^k_{i,j-1})/2h_2)^2}}\Big) \\ & + \frac{\mu}{h_2^2}\triangle^y_-\Big(\frac{\triangle^y_+ \phi^{k+1}_{ij}}{\sqrt{((\phi^k_{i+1,j} - \phi^k_{i-1,j})/2h_1)^2 + (\triangle^y_+ \phi^k_{ij}/h_2)^2}}\Big) \\ & - \nu - \lambda_1(z_{ij} - c_1(\phi^k))^2 + \lambda_2(z_{ij} - c_2(\phi^k))^2\Big]. \end{aligned}$$

For $h_1 = h_2 = h = 1$ which has been mostly used in this thesis for our experiments we

have

$$\frac{\phi_{ij}^{k+1}-\phi_{ij}^{k}}{\triangle t}=\delta_\epsilon(\phi_{ij}^k)\Big[\mu\Big(\frac{\triangle_+^x\phi_{ij}^{k+1}}{\sqrt{(\triangle_+^x\phi_{ij}^k)^2+((\phi_{i,j+1}^k-\phi_{i,j-1}^k)/2)^2}}$$
$$-\frac{\triangle_+^x\phi_{i-1,j}^{k+1}}{\sqrt{(\triangle_+^x\phi_{i-1,j}^k)^2+((\phi_{i-1,j+1}^k-\phi_{i-1,j-1}^k)/2)^2}}\Big)$$
$$+\mu\Big(\frac{\triangle_+^y\phi_{ij}^{k+1}}{\sqrt{((\phi_{i+1,j}^k-\phi_{i-1,j}^k)/2)^2+(\triangle_+^y\phi_{ij}^k)^2}}$$
$$-\frac{\triangle_+^y\phi_{i,j-1}^{k+1}}{\sqrt{((\phi_{i+1,j-1}^k-\phi_{i-1,j-1}^k)/2)^2+(\triangle_+^y\phi_{i,j-1}^k)^2}}\Big)$$
$$-\nu-\lambda_1(u_{0ij}-c_1(\phi^k))^2+\lambda_2(u_{0ij}-c_2(\phi^k))^2\Big].$$

$$\Rightarrow\phi_{ij}^{k+1}=\phi_{ij}^k+\triangle t\delta_\epsilon(\phi_{ij}^k)\Big[\mu\Big(\frac{\phi_{i+1,j}^{k+1}-\phi_{i,j}^{k+1}}{\sqrt{(\triangle_+^x\phi_{ij}^k)^2+((\phi_{i,j+1}^k-\phi_{i,j-1}^k)/2)^2}}$$
$$-\frac{\phi_{i,j}^{k+1}-\phi_{i-1,j}^{k+1}}{\sqrt{(\triangle_+^x\phi_{i-1,j}^k)^2+((\phi_{i-1,j+1}^k-\phi_{i-1,j-1}^k)/2)^2}}\Big)$$
$$+\mu\Big(\frac{\phi_{i+1,j}^{k+1}-\phi_{ij}^{k+1}}{\sqrt{((\phi_{i+1,j}^k-\phi_{i-1,j}^k)/2)^2+(\triangle_+^y\phi_{ij}^k)^2}}$$
$$-\frac{\phi_{ij}^{k+1}-\phi_{i,j-1}^{k+1}}{\sqrt{((\phi_{i+1,j-1}^k-\phi_{i-1,j-1}^k)/2)^2+(\triangle_+^y\phi_{i,j-1}^k)^2}}\Big)$$
$$-\nu-\lambda_1(u_{0ij}-c_1(\phi^k))^2+\lambda_2(u_{0ij}-c_2(\phi^k))^2\Big].$$

Let us denote the coefficients of $\phi_{i+1,j}^{k+1}$, $\phi_{i-1,j}^{k+1}$, $\phi_{i,j+1}^{k+1}$, $\phi_{i,j-1}^{k+1}$ by D_1, D_2, D_3, D_4 respectively, we get the following system of linear equations

$$\begin{aligned}&\phi_{ij}^{k+1}\Big[1+\mu\delta_\epsilon(\phi_{ij}^k)(D_1+D_2+D_3+D_4)\Big]\\&=\phi_{ij}^k+\triangle t\delta_\epsilon(\phi_{ij}^k)\Big[\mu(D_1\phi_{i+1,j}^{k+1}+D_2\phi_{i-1,j}^{k+1}+D_3\phi_{i,j+1}^{k+1}+D_4\phi_{i,j-1}^{k+1})\\&-\nu-\lambda_1(u_{0ij}-c_1(\phi^k))^2+\lambda_2(u_{0ij}-c_2(\phi^k))^2\Big]\end{aligned}\tag{3.31}$$

$$\begin{aligned}\Rightarrow\ \phi_{ij}^{k+1}=&\Big[1+\mu\delta_\epsilon(\phi_{ij}^k)(D_1+D_2+D_3+D_4)\Big]^{-1}\Big\{\phi_{ij}^k+\triangle t\delta_\epsilon(\phi_{ij}^k)\\&\Big[\mu(D_1\phi_{i+1,j}^{k+1}+D_2\phi_{i-1,j}^{k+1}+D_3\phi_{i,j+1}^{k+1}+D_4\phi_{i,j-1}^{k+1})-\nu-\\&\lambda_1(u_{0ij}-c_1(\phi^k))^2+\lambda_2(u_{0ij}-c_2(\phi^k))^2\Big]\Big\}.\end{aligned}\tag{3.32}$$

This linear system of equations can be solved by using any iterative method.

To prevent the level set function becoming too steep or flat a standard procedure is to reinitialize to the signed distance function. Only for a few numerical results we can apply the reinitialization, following the discussion in Section §2.6.4 by solving the following equation [155, 121]:

$$\begin{cases} \dfrac{\partial \xi}{\partial t} = \operatorname{sgn}(\phi(t))(1 - |\nabla \xi|), \\ \xi(0,t) = \phi(t), \end{cases} \tag{3.33}$$

where $\phi(t,.)$ is the solution ϕ at time t, see [37] for more details. Finally, the Chan-Vese algorithm is:

Algorithm 1 Chan-Vese (CV) algorithm for 2-phase image segmentation
CV: $\phi^k \leftarrow CV(\phi^k, u_0, \lambda_1, \lambda_2, \beta, \Delta t, maxit, tol)$.

Initialization: Given $\phi^0_{i,j} = \phi_{i,j,0}$,
for $k = 1 : maxit$ **do**
Compute c_1 and c_2 using equation (3.29)
Solve the PDE in equation (3.29), to update $\phi^k_{i,j}$ computing (3.32)
If $\|\phi^k - \phi^{k-1}\| < tol$ set $\phi^k_{i,j} \leftarrow \phi^{k-1}_{i,j}$, Break;
$\phi^k_{i,j} \leftarrow \phi^{k-1}_{i,j}$, re-initialize ϕ, by solving equation (3.33)

end for

The nonlinear PDE (3.29) was solved with a semi implicit (SI) method, which is unconditionally stable [37, 172] and can also be solved numerically using other similar finite differences schemes. In the case of higher dimensions ($n \geq 2$), with the same arguments as in Subsection §2.9.6, ordering the pixels in such a way that in the i-th row all non-vanishing elements of the system matrix can be found within the positions $[i, i - m_1]$ to $[i, i + m_2]$ is not possible and the typical iterative algorithms such as the Jacobi, Gauss-Seidel, etc. methods have another limitation for large time steps (such as a large condition number) which causes slow convergence.

As discussed in Subsection §2.9.6, the AOS method proposed by J. Weickert [106, 175] for non-linear diffusion filters can be a great way to speed up convergence, especially for higher dimensions ($n \geq 2$). The method is numerically stable for large time steps and converges in fewer iterations. This idea was extended to the Chan-Vese model by M. Jeon [80]. The AOS scheme splits the n-dimensional spatial operator into a sum of n one-dimensional space operators. The update of each point involves only two neighbours in each dimension, thus as result we get a tridiagonal system matrixes. A very fast and parallelizable algorithm such as Thomas algorithm [47] can be used to solve this tridiagonal system.

Let us consider the PDE from equation (3.29)

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\left[\mu\nabla\cdot(\frac{\nabla\phi}{|\nabla\phi|}) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\right]. \tag{3.34}$$

Following the notation in Subsection §2.9.6 here g is represented by the term $\frac{1}{|\nabla\phi|}$. To avoid a singularity of $\frac{1}{|\nabla\phi|}$ we can slightly perturb it by a small β, such that $g = \frac{1}{|\nabla\phi|_\beta} = \frac{1}{\sqrt{\phi_x^2+\phi_y^2+\beta}}$. Let k and i represent time and spatial indices, respectively. At grid point i, the 1-dimensional semi implicit discretization of (3.34) with spatial step $h = 1$ is:

$$\frac{\phi_i^{k+1} - \phi_i^k}{\triangle t} = \delta_\epsilon(\phi_i^k)\left(\frac{1}{|\triangle_+^x \phi_i^k|}(\phi_{i+1}^{k+1} - \phi_i^{k+1}) - \frac{1}{|\triangle_+^x \phi_{i-1}^k|}(\phi_i^{k+1} - \phi_{i-1}^{k+1}) + f_i\right), \tag{3.35}$$

where $f_i = [-\nu - \lambda_1(u_{0i} - c_1)^2 + \lambda(u_{0i} - c_2)^2]$. Denoting

$$D_1 = \frac{1}{|\triangle_+^x \phi_i^k|} \text{ and } D_2 = \frac{1}{|\triangle_+^x \phi_{i-1}^k|},$$

equation (3.35) becomes

$$\phi_i^{k+1} = \phi_i^k + \triangle t(b\phi_{i+1}^{k+1} - a\phi_i^{k+1} + c\phi_{i-1}^{k+1} + f_i). \tag{3.36}$$

where

$$\begin{aligned} a &= \delta_\epsilon(\phi_i^k)D_1, \\ b &= \delta_\epsilon(\phi_i^k)(D_1 + D_2), \\ c &= \delta_\epsilon(\phi_i^k)D_2. \end{aligned}$$

Solving the problem in the x-and y- directions with a double time step, one can get the solution by averaging the two separate solutions, say ϕ_1 and ϕ_2

$$\phi = \frac{1}{2}(\phi_1 + \phi_2).$$

Experimental results in the Chan and Vese paper [37] notice that the model has the tendency to compute a global minimizer. The existence of a global minimum for Chan-Vese was provided by Chan, Esedo$\bar{g}$lu and Nikolova in [31]. Also, for binary image denoising where smoothing of geometric shapes is relevant; the paper shows the existence of a global minimum for ROF model, see Subsection §3.4.11.

3.4.9 Li-Xu-Gui-Fox Level-set Method without Re-initialization

Since a level set function is not unique away from a boundary, a re-initialization might be required that forces the level set function to be close to a signed distance function. There are two different ways to force the level set to be a signed distance function, either by solving a re-initialization equation [121], similar to equation 3.33, or by incorpora-

tion of a functional into the minimization problem [92] and avoid the re-initialization. The first idea introduced in [121] restarts the level set with a costly re-initialization procedure, while the variational formulation consists of an internal energy term that penalizes the deviation of the level set function from a signed distance function.

The new variational model proposed by Li-Xu-Gui-Fox [92] improving the Chan-Vese model by forcing the level set function to be close to a signed distance function formulation is the following

$$\min_{\phi} \mathcal{E}(\phi) = \mu P(\phi) + \alpha L_g(\phi) + \nu A_g(\phi) \tag{3.37}$$

where $\mu > 0$, $\alpha > 0$ and ν are constants, consists of an internal energy term $\mu P(\phi)$ that penalizes the deviation of the level set function from a signed distance function, and an external energy term $\mathcal{E}_m(\phi) = \alpha L_g(\phi) + \nu A_g(\phi)$ that drives the motion of the zero level set toward the desired image features, such as object boundaries. The terms $P(\phi)$, $L_g(\phi)$ and $A_g(\phi)$ are defined by

$$\begin{aligned} P(\phi) &= \int_\Omega \frac{1}{2}(|\nabla\phi| - 1)^2 dxdy, \\ L_g(\phi) &= \int_\Omega g\delta(\phi)|\nabla\phi| dxdy, \\ A_g(\phi) &= \int_\Omega gH(-\phi) dxdy, \end{aligned}$$

where $g = \dfrac{1}{1 + |\nabla G_\sigma * u_0(x,y)|^2}$ for a given image $u_0(x,y)$.

By calculus of variations, the first variation of the functional $\mathcal{E}(\phi)$ in (3.37) can be written as

$$\frac{\partial \mathcal{E}}{\partial \phi} = -\mu[\Delta\phi - \mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})] - \alpha\delta(\phi)\mathrm{div}(g\frac{\nabla\phi}{|\nabla\phi|}) - \nu g\delta(\phi),$$

where Δ is the Laplacian operator. Therefore, the Euler-Lagrange equation $\dfrac{\partial \mathcal{E}}{\partial \phi} = 0$ for the function ϕ is

$$-\mu[\Delta\phi - \mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})] - \alpha\delta(\phi)\mathrm{div}(g\frac{\nabla\phi}{|\nabla\phi|}) - \nu g\delta(\phi) = 0. \tag{3.38}$$

The steepest descent process for minimization of the functional $\mathcal{E}$ is the following gradient flow:

$$\frac{\partial \phi}{\partial t} = \mu[\Delta\phi - \mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})] + \alpha\delta(\phi)\mathrm{div}(g\frac{\nabla\phi}{|\nabla\phi|}) + \nu g\delta(\phi)$$

or

$$\frac{\partial \phi}{\partial t} = \mu[\Delta\phi - \mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})] + \alpha\delta(\phi)[g\,\mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|}) + \nabla g\frac{\nabla\phi}{|\nabla\phi|}] + \nu g\delta(\phi).$$

Hence the level set evolution can be easily implemented by a finite difference scheme and is computationally efficient. This model can be initialised with general functions which are easier to use in practice than the widely used signed distance function.

3.4.10 Piecewise Smooth Segmentation Model by Li-Kao-Gore-Ding

The main difference of the Li-Kao-Gore-Ding model [90, 91] in comparison with Chan-Vese model [37] is that instead of two piecewise constants c_1 and c_2 they propose two piecewise smooth functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ to fit the image intensities near each point $\mathbf{x}$ in a local way. The fitting energy of the model is

$$\begin{aligned}\mathcal{E}_{\mathbf{x}}^{LBF_{Fit}}(\Gamma, f_1(\mathbf{x}), f_2(\mathbf{x})) = \lambda_1 \int_{in(\Gamma)} G_\sigma(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - f_1(\mathbf{x})|^2 d\mathbf{y} + \\ \lambda_2 \int_{out(\Gamma)} G_\sigma(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - f_2(\mathbf{x})|^2 d\mathbf{y},\end{aligned} \tag{3.39}$$

where λ_1 and λ_2 are positive constants, and G_σ is a Gaussian kernel such that $G_\sigma(\mathbf{u}) = \frac{1}{(2\pi)^{n/2}\sigma^n} e^{-|u|^2/2\sigma^2}$, with $\sigma > 0$.

Recall from Subsection §2.7.3, $(f * g)(\mathbf{x}) = \int_{\mathbf{R}^d} f(\mathbf{y}) g(\mathbf{x}-\mathbf{y})\, d\mathbf{y} = \int_{\mathbf{R}^d} f(\mathbf{x}-\mathbf{y}) g(\mathbf{y})\, d\mathbf{y}$ and that the Gaussian kernel $G_\sigma(\mathbf{x}-\mathbf{y})$ takes large value at the points $\mathbf{y}$ near the center point $\mathbf{x}$ and decrease to 0 when $\mathbf{y}$ goes further away. Size of the kernel G_σ controls the intensity $I(\mathbf{y})$ that is effectively involved in the above fitting energy which is effectively zero for $|\mathbf{x}-\mathbf{y}| > 3\sigma$. Moreover it is necessary to emphasize that the fitting energy defined by equation (3.39) is a weighted mean square error of the approximation of the image intensities $I(\mathbf{y})$ by $G_\sigma(\mathbf{x}-\mathbf{y})$.

The fitting energy in equation (3.39) can be rewritten in terms of the level set function as follows

$$\begin{aligned}\mathcal{E}^{LBF_{Fit}}(\phi(\mathbf{x}), f_1(\mathbf{x}), f_2(\mathbf{x})) = \int_\Omega \mathcal{E}_{\mathbf{x}}^{LBF_{Fit}}(\Gamma, f_1(\mathbf{x}), f_2(\mathbf{x})) = \\ \lambda_1 \int \Big[\int G_\sigma(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - f_1(\mathbf{x})|^2 H_\epsilon(\phi(\mathbf{y}))d\mathbf{y} \Big] d\mathbf{x} + \\ \lambda_2 \int \Big[\int G_\sigma(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - f_2(\mathbf{x})|^2 (1 - H_\epsilon(\phi(\mathbf{y})))d\mathbf{y} \Big] d\mathbf{x},\end{aligned}$$

where H_ϵ is the regularized Heaviside function.

The minimization energy proposed by Li-Kao-Gore-Ding model [90, 91] consists of three terms: *i*) the length term, *ii*) fitting term, and *iii*) a regularization term (same as in §3.4.9). The minimization energy in terms of the zero level set is given by

$$\mathcal{F}^{LBF}(\phi(\mathbf{x}), f_1, f_2) = \nu \int_\Omega \delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})| d\mathbf{x} + \mathcal{E}^{LBF_{Fit}}(\phi(\mathbf{x}), f_1, f_2) + \mu \int_\Omega \frac{1}{2}(|\nabla\phi(\mathbf{x})| - 1)^2 d\mathbf{x},$$

where μ and ν are nonnegative constants. By calculus of variations, it can be shown that by keeping ϕ fixed we have

$$\int G_\sigma(\mathbf{x}-\mathbf{y})(I(\mathbf{y}) - f_i(\mathbf{x}))M_i^\epsilon(\phi(\mathbf{y}))d\mathbf{y} = 0, \text{ i=1,2,}$$

where $M_1^\epsilon(\phi(\mathbf{y}) = H_\epsilon(\phi(\mathbf{y}))$ and $M_2^\epsilon(\phi(\mathbf{y}) = 1 - H_\epsilon(\phi(\mathbf{y}))$, or

$$\int G_\sigma(\mathbf{x}-\mathbf{y})I(\mathbf{y})M_i^\epsilon(\phi(\mathbf{y}))d\mathbf{y} = \int G_\sigma(\mathbf{x}-\mathbf{y})f_i(\mathbf{x})M_i^\epsilon(\phi(\mathbf{y}))d\mathbf{y},\ \text{i=1,2,}$$

$\Downarrow$

$$\int G_\sigma(\mathbf{x}-\mathbf{y})I(\mathbf{y})M_i^\epsilon(\phi(\mathbf{y}))d\mathbf{y} = f_i(\mathbf{x})\int G_\sigma(\mathbf{x}-\mathbf{y})M_i^\epsilon(\phi(\mathbf{y}))d\mathbf{y},\ \text{i=1,2,}$$

$\Downarrow$

$$f_i(\mathbf{x}) = \frac{\int G_\sigma(\mathbf{x}-\mathbf{y})I(\mathbf{y})M_i^\epsilon(\phi(\mathbf{y}))d\mathbf{y}}{\int G_\sigma(\mathbf{x}-\mathbf{y})M_i^\epsilon(\phi(\mathbf{y})d\mathbf{y}},\ \text{i=1,2,}$$

and in this way we update $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$

$$f_1(\mathbf{x}) = \frac{G_\sigma(\mathbf{x}) * [H_\epsilon(\phi(\mathbf{x}))I(\mathbf{x})]}{G_\sigma(\mathbf{x}) * H_\epsilon(\phi(\mathbf{x}))}, \quad f_2(\mathbf{x}) = \frac{G_\sigma(\mathbf{x}) * [(1 - H_\epsilon(\phi(\mathbf{x}))I(\mathbf{x})]}{G_\sigma(\mathbf{x}) * (1 - H_\epsilon(\phi(\mathbf{x})))}.$$

Keeping $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ fixed one can get

$$\frac{\partial\phi(\mathbf{x})}{\partial t} = -\delta_\epsilon(\phi(\mathbf{x}))(\lambda_1 e_1 - \lambda_2 e_2) + \nu\delta_\epsilon(\phi(\mathbf{x}))\,\mathrm{div}(\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}) + \mu(\nabla^2\phi(\mathbf{x}) - \mathrm{div}(\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|})). \tag{3.40}$$

where e_1, e_2 are functions calculated as follows

$$e_1 = \int_\Omega G_\sigma(\mathbf{y}-\mathbf{x})|I(\mathbf{x}) - f_1(\mathbf{y})|^2 d\mathbf{y},\ e_2 = \int_\Omega G_\sigma(\mathbf{y}-\mathbf{x})|I(\mathbf{x}) - f_2(\mathbf{y})|^2 d\mathbf{y}.$$

An example that shows the difference between this model and Chan-Vese will be shown in Section §6.5.

3.4.11 Global Minimization of the Active Contour Model

To avoid the existence of local minima in the active contour energy, which makes the method depend on the initial guess, T. F. Chan *et al.* [31] proposed to solve this problem by determining a global minimum. In their paper, first is proposed a denoising model for binary image which removes the geometric noise in a given shape and the second model is the active contours without edges model of Chan and Vese. Later a new numerical way to solve the active contour propagation through a dual formulation was proposed by Bresson *et al.* [23, 22]. Assuming that the image domain Ω is a bounded subset of $\mathbb{R}^n$ the 2-phase image segmentation model Chan-Vese model (3.20) is written

$$\begin{aligned}\min_{\phi,c_1,c_2} F^{CV}(\phi,c_1,c_2) = \mu\int_\Omega |\nabla H(\phi)|d\mathbf{x} + \eta\int_\Omega H(\phi)d\mathbf{x} + \\ \lambda_1\int_\Omega |u_0(\mathbf{x}) - c_1|^2 H(\phi)d\mathbf{x} + \lambda_2\int_\Omega |u_0(\mathbf{x}) - c_2|^2(1 - H(\phi))d\mathbf{x}.\end{aligned} \tag{3.41}$$

This optimization problem can be interpreted as the best approximation in the L^2 sense to the given image $u_0(\mathbf{x})$ among all functions that take only two values. This minimization problem is non-convex because the minimization is carried out over a

nonconvex set of functions [31]. Hence the optimization problem can have local minima, and can have more than one minimizer. The approximation of the solution in a two-step scheme at the first step computes c_1 and c_2, and at the second step updates the curve shape with the gradient descent equation

$$\frac{\partial \phi}{\partial t} = H'_\epsilon(\phi)\left[\mu\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \underbrace{(\eta + \lambda_1(u_0 - c_1)^2 - \lambda_2(u_0 - c_2)^2)}_{r(\mathbf{x})}\right]$$

$$\Rightarrow \frac{\partial \phi}{\partial t} = H'_\epsilon(\phi)\left[\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda r(\mathbf{x})\right], \tag{3.42}$$

with $\lambda = 1/\mu$.

The two-phase model depends on the level set function only through the term $H(\phi)$ representing a parametrization of binary functions. So the minimization is thus a minimization over binary functions. The choice of the regularization $H_\epsilon(\phi)$ of $H(\phi)$ is a crucial ingredient of the CV algorithm.

The function $H_\epsilon(\phi)$ is a regularization of the Heaviside function which in [37] the authors chose as being a non-compactly supported, smooth and strictly monotone approximation. As a result, the steady state solution of the gradient flow (3.42) is the same as:

$$\frac{\partial \phi}{\partial t} = \left[\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda r(\mathbf{x})\right], \tag{3.43}$$

where $H'_\epsilon(\phi)$ is simply omitted. This equation (3.43) is the gradient descent equation of the following convex energy:

$$\int_\Omega |\nabla\phi| + \lambda\int_\Omega r(\mathbf{x})\phi d\mathbf{x}. \tag{3.44}$$

This energy is homogeneous of degree 1 in ϕ, so as a result in general it does not have a minimizer if we do not restrict the minimization to ϕ such as $0 \le \phi \le 1$, $\forall(\mathbf{x}) \in \Omega$. With $0 \le \phi \le 1$ we have the following minimization problem :

$$\min_{0\le\phi(\mathbf{x})\le 1} F^{CVG}(\phi, c_1, c_2), \tag{3.45}$$

$$F^{CVG}(\phi, c_1, c_2) = \left(\int_\Omega |\nabla\phi| d\mathbf{x} + \lambda\int_\Omega r(\mathbf{x})\phi d\mathbf{x}\right). \tag{3.46}$$

This minimization, as shown in [31], leads to the global minimizer from the following theorem:

Theorem 3.4.1 *For any given fixed $c_1, c_2 \in \mathbb{R}$, a global minimizer for $F^{CVG}(u, c_1, c_2)$*

can be found by carrying out the following convex minimization

$$\min_{0\leq u(\mathbf{x})\leq 1} \underbrace{\left(\int_\Omega |\nabla u| d\mathbf{x} + \lambda \int_\Omega r(\mathbf{x}) u d\mathbf{x} \right)}_{F^{CVG}(u,c_1,c_2)}. \tag{3.47}$$

and then setting $\Gamma = \{x : u(x) > \mu\}$ *for a.e.* $\mu \in [0,1]$.

In comparison with CV model, in this theorem the minimization of (3.47) removes the non-convex constraint of being binary and instead we minimize over functions that are allowed to take intermediate values. The minimization (3.47) is changed into an unconstrained minimization problem according to the following theorem:

Theorem 3.4.2 *Let* $r(\mathbf{x}, c_1, c_2) \in L^\infty(\Omega)$*, for any given* $c_1, c_2 \in R$*, and* $\lambda \in R^+$*. Then the convex constrained minimization problem (3.47) has the same set of minimizers as the following convex, unconstrained minimization problem*

$$\min_u \left(\int_\Omega |\nabla u| d\mathbf{x} + \lambda \int_\Omega (r(\mathbf{x}, c_1, c_2) u + \alpha \nu(u)) d\mathbf{x} \right), \tag{3.48}$$

where $\nu(\zeta) := \max\{0, 2|\zeta - \frac{1}{2}| - 1\}$ *is an exact penalty function, provided that the constant* $\alpha > \frac{\lambda}{2} \|r(\mathbf{x})\|_{L^\infty(\Omega)}$.

The proof and further details can be found in [31].

The energy $F^{CVG}(u, c_1, c_2)$ is convex but not strictly convex, which means that $F^{CVG}(u, c_1, c_2)$ does not possess local minima that are not global minima. Hence any minimizer of $F^{CVG}(u, c_1, c_2)$ is a global minimizer. The Euler Lagrange equation is given by

$$\frac{\partial u}{\partial t} = \left[\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \lambda r(\mathbf{x}, c_1, c_2) \right] - \alpha \nu'_{\epsilon_2}(u),$$

where ν'_{ϵ_2} is the regularized version of ν' with $\nu_{\epsilon_2}(\zeta)$ is given by [23]:

$$\nu_{\epsilon_2}(\zeta) = \begin{cases} -\zeta & \text{if} \quad \zeta < -\epsilon_2/\sqrt{2}, \\ (1+\sqrt{2})\zeta - \sqrt{\tan^2(3\pi/8)\zeta^2 - (\zeta - \epsilon_2)^2} & \text{if} \quad -\epsilon_2/\sqrt{2} \leq \zeta < \epsilon_2, \\ 0 & \text{if} \quad \epsilon_2 \leq \zeta < 1 - \epsilon_2, \\ (1+\sqrt{2})\zeta - \sqrt{\tan^2(3\pi/8)\zeta^2 - (\zeta - 1 + \epsilon_2)^2} & \text{if} \quad 1 - \epsilon_2 \leq \zeta < 1 + \epsilon_2/\sqrt{2}, \\ \zeta - 1 & \text{if} \quad \zeta \geq 1 + \epsilon_2/\sqrt{2}. \end{cases}$$

An edge enhancement energy which unifies the snake and active contour without edges models was proposed in [23, 22]. The difference with energy (3.47) is made by a function $g(\mathbf{x})$ which weights the TV-norm. The new energy functional for the weighted TV is given in the form:

$$\min_{0\leq u(\mathbf{x})\leq 1} \left(\int_\Omega g(\mathbf{x}) |\nabla u| d\mathbf{x} + \lambda \int_\Omega r(\mathbf{x}, c_1, c_2) u d\mathbf{x} \right), \tag{3.49}$$

where $g(\mathbf{x})$ is an edge detector function and is defined as in (3.13) and $r(\mathbf{x}) = (\eta + \lambda_1(u_0 - c_1)^2 - \lambda_2(u_0 - c_2)^2)$. Based on dual formulation of the TV-norm idea, the algorithm introduced on [23, 22] improves the old work [31] by defining a fast segmentation algorithm. Bresson *et al.* [23] rewrites the variational problem (3.48) as

$$\min_{u,v} \left(\int_\Omega g(\mathbf{x})|\nabla u| d\mathbf{x} + \frac{1}{2\theta}||u - v||^2_{L^2} + \lambda \int_\Omega (r(\mathbf{x}, c_1, c_2)v + \alpha\nu(v)) d\mathbf{x} \right), \tag{3.50}$$

where the parameter $\theta > 0$ is chosen to be small.

Since the functional (3.48) is convex, its minimizer can be computed by minimizing separately u and v and the computation will be iterated until convergence. Thus, the following minimization problems are considered:

- v being fixed, we search for u as a solution of:

$$\min_u \left(\int_\Omega g(\mathbf{x})|\nabla u| d\mathbf{x} + \frac{1}{2\theta}||u - v||^2_{L^2} \right), \tag{3.51}$$

- u being fixed, we search for v as a solution of:

$$\min_v \left(\frac{1}{2\theta}||u - v||^2_{L^2} + \lambda \int_\Omega (r(\mathbf{x}, c_1, c_2)v + \alpha\nu(v)) d\mathbf{x} \right). \tag{3.52}$$

Proposition 3.4.3 *The solution of (3.51) is given by*

$$u = v - \theta \operatorname{div} p, \tag{3.53}$$

where $p = (p^1, p^2)$ is given by

$$g(\mathbf{x})\nabla(\theta \operatorname{div} p - v) - |\nabla(\theta \operatorname{div} p - v)|p = 0, \tag{3.54}$$

which can be solved by a fixed point method:

$$\begin{aligned} p^0 &= 0, \\ p^{n+1} &= \frac{p^n + \triangle t \nabla(\operatorname{div}(p^n) - v/\theta)}{1 + \frac{\triangle t}{g(\mathbf{x})}|\nabla(\operatorname{div}(p^n) - v/\theta)|}. \end{aligned} \tag{3.55}$$

Proposition 3.4.4 *The solution of (3.52) is given by*

$$v = \min\{\max\{u(\mathbf{x}) - \theta\lambda r(\mathbf{x}, c_1, c_2), 0\}, 1\}. \tag{3.56}$$

See proof and experimental results in [23, 22].

3.5 Conclusion

In this chapter we mainly discussed some variational models, starting with image denoising as a model in image processing which can easily be understood by the readers,

and continued with image segmentation. We also introduced some numerical methods used for solving the parabolic PDE arising from the minimization of these models. Since mainly we will deal with segmentation in this thesis we tried to give a short overview of a few models in segmentation such as Geodesic Active Contours model [28], Chan-Vese method [37], Li-Xu-Gui-Fox [92] model without re-initialization, and a global minimazation model introduced by T. F. Chan *et al.* [31].

The Geodesic Active Contours uses local information of the boundary and curvature to detect the objects. This model can fail when dealing with fuzzy edges and can be strongly affected by heavy noise. The Chan-Vese method depends on the image information derived from homogenous regions, therefore gets better results in comparison with Geodesic Active Contours in fuzzy and discrete cases, by detecting objects in images with or without noise and whose boundaries are not defined by gradient. The drawback of the Chan-Vese model of getting stuck in a local minima can be avoided with a cheap reinitialization as in Li-Xu-Gui-Fox [92]. A fast global minima can be found using the numerical results in [23, 22] by restricting the image intensity to be between 0 and 1.

Solving the parabolic PDEs with an explicit scheme is popular due to the convergence properties but requires very small time steps and therefore is slow. Implicit schemes are stable but the system matrix has a large condition number for large time steps which causes slow convergence for iterative methods. The AOS scheme introduced in Subsection 2.9.6 is unconditionally stable implicit schemes and is very efficient.

Chapter 4

A Restarted Iterative Homotopy Analysis Method for Two Nonlinear Models from Image Processing

Total variation minimization based nonlinear models have been proven to be very useful and successful in image processing. A lot of effort has been devoted overcoming the nonlinearity of the model and at the same time increasing the speed is one of the remaining challenges. In this chapter, we study a restarted homotopy analysis method submitted for publication in [59, 130] in collaboration with my supervisor Prof. Ke Chen and Dr. Behzad Ghanbari. The method modifies the existing homotopy analysis method [94] introduced in Subsection §2.9.7 and makes it valid for hard nonlinear problems with applications in image processing. The algorithm improves the computational efficiency and we will show by experiments that this method demonstrates great potential for recovering the noise and operates with great speed in both image denoising and image segmentation models.

4.1 Introduction

Due to the inevitable presence of noise while acquiring images, a pre-processing step is the first common process in image analysis. Various denoising techniques are proposed for removing noise and preserving important image information such as edges [165]. Denoising via linear filters normally do not perform satisfactorily, since both noise and edges contain high frequencies, while nonlinear denoising models have been applied and found to be successful. Rudin, Osher and Fatemi [135, 136] first introduced the TV norm as an edge preserving model. This model is a successful tool not only for image restoration (denoising) but for edge enhancement (segmentation) as well. Among image segmentation models Chan-Vese active contours without edges, which is a TV based method [115, 37, 39, 166], makes use of image intensities considering a certain

homogeneity to guide the motion of active contours. Details about these two models have been already given in Chapter 3. High order effective models [24, 186] improving on TV have been proposed, but their nonlinear PDEs bring some implementation and speed difficulties [178].

The Euler-Lagrange equation associated with the TV functional for those models is a nonlinear PDE. Due to the difficulty (even impossibility) of obtaining analytic solutions for the nonlinear PDEs it is easier to get a numerical approximation of a given nonlinear problem. Using an implicit method as a solver brings the dependence of convergence on certain properties of the matrix of the system after digitization, complicated implementation or requirement of large storage. On the other hand, explicit methods (such as the easily implemented time marching model) requires the time step to be chosen sufficiently small to get convergence and speeding up would be a challenge.

As detailed in §2.9.7, the HAM attracts with the simple idea of transforming the nonlinear problem into the solution of a sequence of linear equations. In this way the method brings improved accuracy of high order linear approximation as well as fast convergence. Unfortunately, applying it to realistic nonlinear PDE models does not suit due to the analytical difficulties in working out the high order approximations demanded by HAM. Sorting this out and at the same time using HAM's quality were the motivation for our work.

In this chapter we develop a discrete restarted homotopy analysis method (RHAM) which proposes to use the sequence of three linear equations and to restart the HAM process afterwards thereby avoiding the requirements of using high order terms. It will be shown that RHAM has the same or even better efficiency when compared with HAM, while being simple and easy to apply.

We will demonstrate through 2-D and 3-D examples that RHAM is comparable with HAM for solving hard inverse problems such as those encountered in image processing. This method mathematically brings a stable scheme, has the same or improved quality, has a general formulation for nonlinear equations and deals with large 2-D images and 3-D data where speed is crucial. The application of RHAM are for TV based models, which suggests that it may be used for other un-explored hard nonlinear applications as well.

In what follows, in Section §4.2 we first review the HAM implementation and the difficulties of applying it to the classical TV restoration model. In this section we develop the general idea of RHAM motivated by overcoming these difficulties. Then in Section §4.3 the classical TV restoration model and its discrete form are presented. In this section we will develop and present the restarted iterative homotopy analysis method for the model. In Section §4.4, we describe the developed discrete homotopy analysis method for segmentation. In Section §4.5 we present various numerical results obtained from the implementation of RHAM for both denoising restoration and the segmentation models, and then comparisons are made between obtained results and

those of time marching and the additive operator splitting methods.

4.2 Restarted Homotopy Analysis Method

In this section we will show the motivation of introducing a restarted homotopy analysis method (RHAM) as a discrete modified form of HAM introduced in Subsection §2.9.7 for hard nonlinear cases.

Recall from §2.9.7, for a given nonlinear problem $\mathcal{N}[u(\mathbf{x},t)] = 0$ one can use the recurrence formulas (2.132)

$$u_m^{DHAM1}(\mathbf{x},t) = \chi_m u_{m-1}^{DHAM1}(\mathbf{x},t) + \hbar exp(-\theta t)\int_0^t exp(\theta t)\mathcal{H}_1(\tau)R_m[\vec{u}_{m-1}^{DHAM1}]d\tau \quad (4.1)$$

and (2.133)

$$u_m^{DHAM2}(\mathbf{x},t) = \chi_m u_{m-1}^{DHAM2}(\mathbf{x},t) + \frac{\hbar}{1+t}\int_0^t \mathcal{H}_2(\tau)R_m[\vec{u}_{m-1}^{DHAM2}]d\tau \quad (4.2)$$

to successively obtain all $u_m(t)$ for $m \geqslant 1$ and afterwards the Mth order approximation of the form $u_M(\mathbf{x},t) = \sum_{m=0}^{M} u_m(\mathbf{x},t)$, for two different linear operators $\mathcal{L}_1$ and $\mathcal{L}_2$ given by equations (2.130) and (2.131), respectively, and for a given initial approximation $u_0(\mathbf{x},t) = u_0(\mathbf{x})$. To calculate $u_m(t)$ we need to input $R_m[\vec{u}_{m-1}(\mathbf{x})]$ given by equation (2.128) as follows:

$$R_m[\vec{u}_{m-1}(\mathbf{x})] = \frac{1}{(m-1)!}\frac{\partial^{m-1}\mathcal{N}[\varphi(\mathbf{x},t;q)]}{\partial q^{m-1}}\bigg|_{q=0}. \quad (4.3)$$

We find the method attractive due to the high order accuracy offered and the challenge of answering the simple question "Is it easy to apply HAM to realistic nonlinear PDE models particularly in image processing?".

In image processing techniques, such as denoising and image segmentation, we deal with strongly nonlinear PDEs arising from the nonlinear TV term

$$\nabla \cdot \Big(\frac{\nabla u(\mathbf{x})}{|\nabla u(\mathbf{x})|_\beta}\Big).$$

More complicated cases can be found when using high order PDEs, for example the curvature model

$$\alpha\nabla \cdot \left(\frac{\nabla \Phi'(\kappa)}{|\nabla u|_\beta} - \frac{\nabla u \cdot \nabla \Phi'(\kappa)}{(|\nabla u|_\beta)^3}\nabla u\right) \quad (4.4)$$

with κ the curvature of the image and Φ defined either as $\Phi(\kappa) = |\kappa|$, $\Phi(\kappa) = \kappa^2$ or as in [185, 24] as a combination of both. When applying the HAM method we require the derivation of the R_m term in equation (4.3) with respect to q. If we carry on with approximations of order $m = 3, m = 4, \ldots, m = M$ the derivation of the R_m term, even though it is with respect to q and does not bring a higher order than the given PDE, still might require not only calculation but even some difficulties in programming a

few more terms. To avoid all this we can stop at a certain M (depending on the difficulty of the given nonlinear problem, $M = 2$ in our case) and restart with a better approximation

$$\hat{u}_M = u_0^{(1)} + \sum_{r=1}^{R} \sum_{m=1}^{M} u_m^{(r)}. \tag{4.5}$$

This is the rough idea behind the so-called restarted homotopy analysis method which will be given in detail in the following sections for both image denoising and image segmentation.

"Does RHAM work for simple cases such as ODEs?" would be the first question to answer before going to complicated image processing cases. For this reason we tested the idea for simpler problems that were already considered with the HAM method and shown to be successful [21, 93]. In the following we show a great performance of RHAM for these examples in comparison with HAM. The results show that a second order RHAM performs as the 8^{th} or the 10^{th} order HAM.

A Simple Comparison of RHAM and HAM. To compare the RHAM performance with HAM we consider two simple examples. The first example shows how quick the RHAM converges to the right solution. Considering the following nonlinear second-order boundary value problem

$$\begin{cases} u''^3 - 6u(x) - 2x^3 = 0 \\ u(1) = 2, u(2) = \frac{5}{2}. \end{cases}$$

It can be found that the exact solution has the form $u(x) = x + \frac{1}{x}$. Starting with the initial condition $u_0(x) = x^2 - \frac{5}{2}x + \frac{7}{2}$, it can be shown, Figure 4.1, that the RHAM obtained in 4 restarted iterations reaches a good approximation to the exact solution while HAM needs to be 8^{th} order expanded for the same results.

As a second example we consider the first-order boundary value problem

$$\begin{cases} u' - u^2(x) - 1 = 0, \\ u(0) = 0, \end{cases}$$

with exact solution in the form $u(x) = \tanh(x)$. Figure 4.2 shows the result of RHAM obtained in 5 restarted iterations in comparison with high order HAM up to 10^{th} order, for the considered initial guess $u_0(x) = 1 - \frac{1}{1+x}$.

Other examples have been tested and have shown that the RHAM needs only a few re-initialization iterations to reach the same performance as high order HAM.

4.3 A Restarted Iterative Homotopy Analysis Method for Total Variational Denoising

We use this section to carry out a quick revision of the denoising formulation as a variational problem and the formulation of a restarted homotopy analysis method for

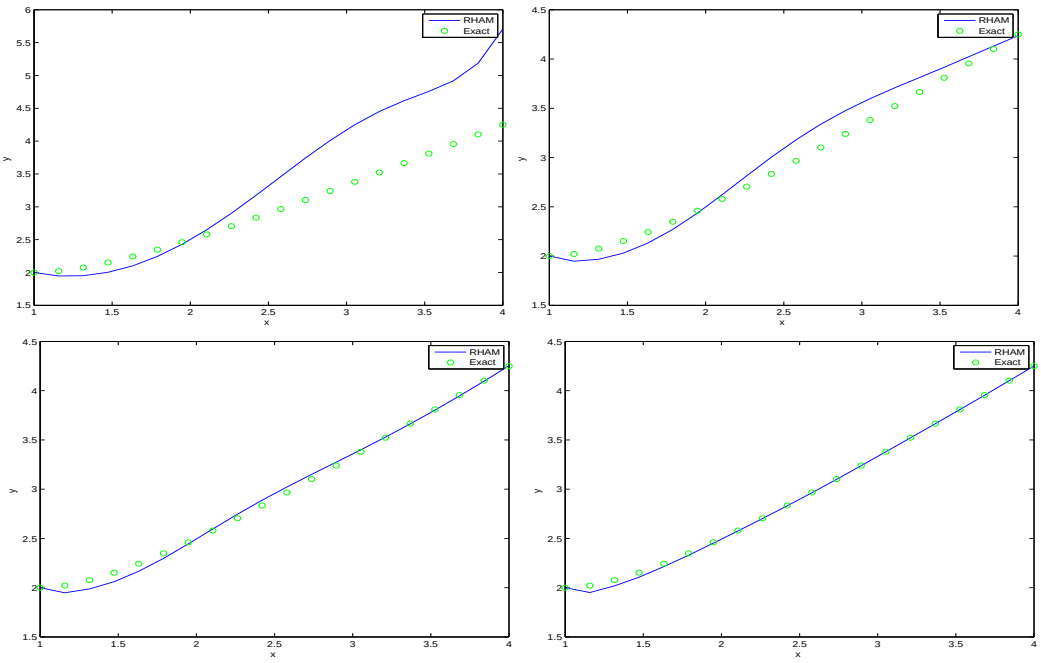


Figure 4.1: From the top left to the bottom right, the evolution of RHAM reaching great accuracy in comparison with the exact solution. The blue line shows the result of RHAM after 1, 2, 3 and 4 re-initialization and in green, the exact solution. The results shown in the last figure are the same as HAM which includes more expanded terms (HAM needs to be 8^{th} order expanded).

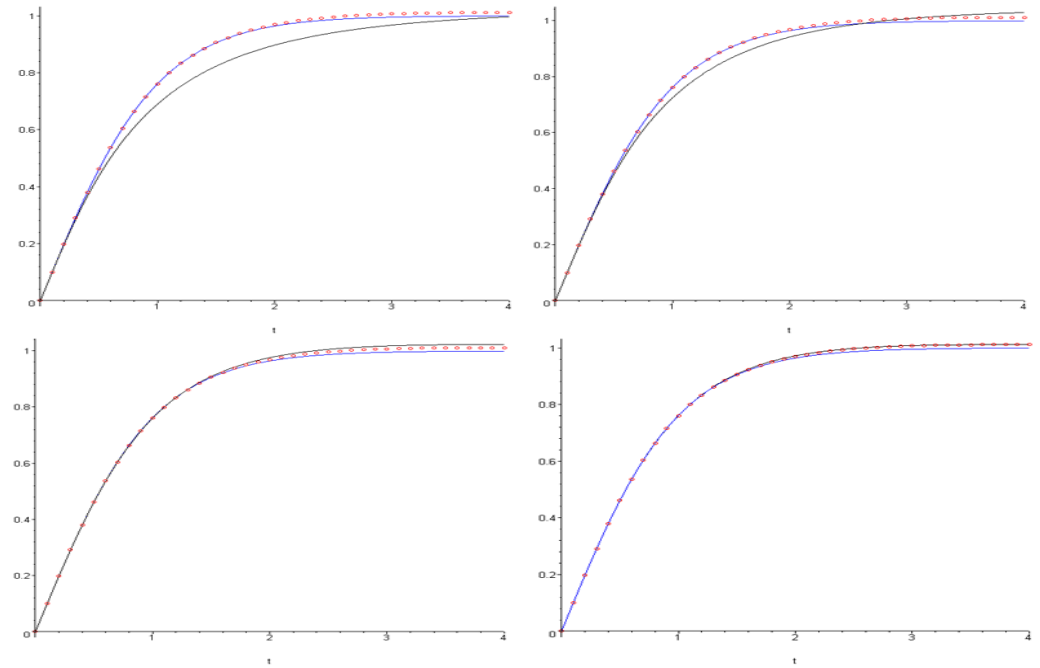


Figure 4.2: In blue the exact solution, red RHAM for $m = 2$ with 5 re-initializations and black HAM. From the top left to the bottom right HAM is increasing in order from 2, to 4, 8 and finally 10^{th}, which reaches the same accuracy as second order RHAM with 5 re-initializations for 10^{th} order HAM.

TV in denoising. The first subject has been covered in Chapter 3 so we do not extend ourselves again with extensive explanations.

4.3.1 The Total Variation Model

Same to Subsection §3.3.1 we assume that $u_0(\mathbf{x})$ is a given image (known) in Ω, which is a bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, contaminated with unknown additive Gaussian noise. We want to construct $u(\mathbf{x})$ (unknown), the desired clean image for the observed image $u_0(\mathbf{x})$, such that $u_0(\mathbf{x}) = u(\mathbf{x}) + \eta(\mathbf{x})$. Following Rudin, Osher and Fatemi [135], the TV denoising model given in equation (3.7) leads to the Euler-Lagrange equation, a nonlinear elliptic partial differential equation with homogeneous Neumann boundary conditions as follows

$$\begin{aligned} \wp\Big(u(\mathbf{x})\Big) = \nabla \cdot \Big(\frac{\nabla u(\mathbf{x})}{|\nabla u(\mathbf{x})|_\beta}\Big) - \lambda\Big(u(\mathbf{x}) - u_0(\mathbf{x})\Big) = 0, \quad \mathbf{x} \in \Omega, \\ \nabla u(\mathbf{x}) \cdot \vec{n} = 0, \quad \mathbf{x} \in \partial\Omega, \end{aligned} \tag{4.6}$$

where $\lambda > 0$ is a tuning parameter, $\vec{n}$ is the unit normal vector exterior to the boundary $\partial\Omega$ and $\beta > 0$ is some small parameter helping to avoid division by zero in the numerical implementation of the term $|\nabla u(\mathbf{x})|$ by a smooth approximation $|\nabla u(\mathbf{x})|_\beta = \sqrt{|\nabla u(\mathbf{x})|^2 + \beta^2}$.

4.3.2 Time Marching Method

For solving the Euler-Lagrange equation (4.6), Rudin, Osher and Fatemi [135] used a parabolic equation as a solution procedure, that means we solve

$$\begin{aligned} \frac{\partial u(\mathbf{x};t)}{\partial t} = \nabla \cdot \Big(\frac{\nabla u(\mathbf{x};t)}{|\nabla u(\mathbf{x};t)|_\beta}\Big) - \lambda\Big(u(\mathbf{x};t) - u_0(\mathbf{x})\Big), \ t > 0, \mathbf{x} \in \Omega \\ \nabla u \cdot \vec{n} = 0, \ \mathbf{x} \in \partial\Omega \end{aligned} \tag{4.7}$$

for a given image $u(\mathbf{x}, 0) = u_0(\mathbf{x})$. For the numerical implementation, we will show the 2-D discretization and time marching realization of the model. Let us assume that the domain $\Omega \in \mathbb{R}^2$ has been split into $N \times M$ cells where the grid points are located at $(x_i = ih_x, y_j = jh_y)$, $i = 1, ..., N$, $j = 1, ..., M$, $t_k = k\Delta t$, where Δt and $k = 1, 2, ...,$ denote the time step and iteration time, respectively. We denote the values of $u(x, y, t)$ at the grid points (x_i, y_j, t_k) by $u^k_{i,j}$ and $u^0_{i,j} = u_0(x_i, y_j)$. For simplicity and without lost of generality we assume that $M = N$, $h_x = h_y = 1$.

In this way, the curvature term can be approximated by

$$\begin{aligned} \kappa(u^k_{i,j}) = \nabla \cdot \Big(\frac{\nabla u}{|\nabla u|_\beta}\Big)\Big|_{i,j} = \Big(\frac{\partial}{\partial x}\big[\frac{u_x}{|\nabla u_\beta|}\big] + \frac{\partial}{\partial y}\big[\frac{u_y}{|\nabla u_\beta|}\big]\Big)\Big|_{i,j} \\ = \Delta^x_- \left(\frac{\Delta^x_+ u^k_{ij}}{\sqrt{(\Delta^x_+ u^k_{ij})^2 + (\Delta^y_+ u^k_{ij})^2 + \beta}}\right) + \Delta^y_- \left(\frac{\Delta^y_+ u^k_{ij}}{\sqrt{(\Delta^y_+ u^k_{ij})^2 + (\Delta^y_+ u^k_{ij})^2 + \beta}}\right) \end{aligned} \tag{4.8}$$

with

$$\Delta^x_{\mp} u_{ij} = \mp(u_{i\mp 1j} - u_{ij})$$

$$\Delta^y_{\mp} u_{ij} = \mp(u_{ij\mp 1} - u_{ij})$$

first order differences and $\wp(u^k_{i,j})$ calculated in the following form

$$\wp(u^k_{i,j}) = \kappa(u^k_{i,j}) - \lambda(u^k_{ij} - u_{0ij}). \tag{4.9}$$

If the time derivative u_t at $(i, j, k\Delta t)$ is approximated by the forward difference, as

$$(u_t)^k_{ij} = \frac{u^{k+1}_{ij} - u^k_{ij}}{\Delta t}$$

then by considering equation (4.9), we obtain

$$\frac{u^{k+1}_{ij} - u^k_{ij}}{\Delta t} = \wp(u^k_{i,j}) \tag{4.10}$$

or

$$u^{k+1}_{ij} = u^k_{ij} + \Delta t \wp(u^k_{i,j}) \ , \tag{4.11}$$

with boundary condition $u^k_{0j} = u^k_{1j}, \quad u^k_{Nj} = u^k_{N-1j}, \quad u^k_{i0} = u^k_{i1}, \quad u^k_{iN} = u^k_{iN-1}$ for $i, j = 1, 2, ..., N$.

The time marching method for solving equation (4.7) is described below in Algorithm 2:

Algorithm 2 Time Marching Method TM: $u^k \leftarrow TM(u^k, u_0, \lambda, maxit, tol)$.

for $iter = 1 : maxit$ **do**

Compute $\wp(u^k_{i,j})$ using (4.8),

Perform TM steps on linear system by (4.11): $u^{k+1}_{i,j} \leftarrow u^k_{i,j} + \Delta t \wp(u^k_{i,j})$

If $\|u^k - u^{k+1}\| < tol$ or $PSNR(u^k) > PSNR(u^{k+1})$, set $u^k_{i,j} \leftarrow u^{k+1}_{i,j}$, Break;

$u^k_{i,j} \leftarrow u^{k+1}_{i,j}$

end for

4.3.3 A Restarted Homotopy Analysis Method for Denoising

Considering the above nonlinear partial differential equation (4.7) we define the nonlinear partial differential equation as

$$\mathcal{N}[u(\mathbf{x}, t)] = \frac{\partial u(\mathbf{x}, t)}{\partial t} - \nabla \cdot \Big(\frac{\nabla u(\mathbf{x}, t)}{|\nabla u(\mathbf{x}, t)|_\beta}\Big) + \lambda\Big(u(\mathbf{x}, t) - u_0(\mathbf{x})\Big) = 0. \tag{4.12}$$

The RHAM can be used to solve the equation by choosing the nonlinear operator $\mathcal{N}[\varphi(\mathbf{x}, t; q)]$ in a straightforward manner

$$\mathcal{N}[\varphi(\mathbf{x}, t; q)] = \frac{\partial \varphi(\mathbf{x}, t; q)}{\partial t} - \nabla \cdot \Big(\frac{\nabla \varphi(\mathbf{x}, t; q)}{|\nabla \varphi(\mathbf{x}, t; q)|_\beta}\Big) + \lambda\Big(\varphi(\mathbf{x}, t; q) - u_0(\mathbf{x})\Big), \tag{4.13}$$

where $0 \leq q \leq 1$.
The two linear operators $\mathcal{L}_1$ and $\mathcal{L}_2$ introduced in Subsection §2.9.7 will be considered for solving the nonlinear partial differential equation (4.7), with initial approximation

$$\varphi(\mathbf{x}, t = 0) = u_0(\mathbf{x}).$$

In what follows, we give the solutions of (4.1) and (4.2) for $m = 1$ and 2.

Keeping the notation u_0 for u_0^{RHAM1} and u_0^{RHAM2}, and u_1 for u_1^{RHAM1} and u_1^{RHAM2} respectively, while applying (4.3) in the following calculation of $R_1[\vec{u}]$ and $R_2[\vec{u}]$ we have

$$R_1[u_0] = \frac{\partial^0 N[u(\mathbf{x}, t; q)]}{\partial^0 q}\bigg|_{q=0} = \mathcal{N}(u_0) = -\wp(u_0), \tag{4.14}$$

$$\begin{aligned} R_2[u_1, u_0] &= \frac{\partial N[\varphi(\mathbf{x}, t; q)]}{\partial q}\bigg|_{q=0} = \frac{\partial}{\partial q}\Big(\frac{\partial \varphi}{\partial t} - \nabla \cdot (\frac{\nabla \varphi}{|\nabla \varphi|_\beta}) + \lambda(\varphi - u_0)\Big)\bigg|_{q=0} \\ &= \Big(\frac{\partial}{\partial t}\frac{\partial \varphi}{\partial q} - \frac{\partial}{\partial x}(\frac{\partial}{\partial q}\frac{\varphi_x}{|\nabla \varphi|_\beta}) - \frac{\partial}{\partial y}(\frac{\partial}{\partial q}\frac{\varphi_y}{|\nabla \varphi|_\beta}) + \frac{\partial(\lambda(\varphi - u_0))}{\partial q}\Big)\bigg|_{q=0}, \end{aligned}$$

$$\begin{aligned} \Rightarrow R_2[u_1, u_0] = u_{1t} - \frac{\partial}{\partial x}\Big(\frac{u_{1x}}{|\nabla u_0|_\beta} - \frac{u_{0x}u_{1x} + u_{0y}u_{1y}}{|\nabla u_0|_\beta^3}u_{0x}\Big) - \\ \frac{\partial}{\partial y}\Big(\frac{u_{1y}}{|\nabla u_0|_\beta} - \frac{u_{0x}u_{1x} + u_{0y}u_{1y}}{|\nabla u_0|_\beta^3}u_{0y}\Big) - \lambda u_1 \end{aligned} \tag{4.15}$$

From equations (4.1) and (4.2), we have

for RHAM1

$$u_1^{RHAM1} = \hbar exp(-\theta t)\int_0^t exp((\theta - 2)\tau)R_1[\vec{u}_0]d\tau = -\hbar\frac{e^{-2t} - e^{-\theta t}}{\theta - 2}\wp(u_0), \tag{4.16}$$

$$u_2^{RHAM1} = u_1^{RHAM1} + \hbar exp(-\theta t)\int_0^t exp((\theta - 2)\tau)R_2[u_1^{RHAM1}, u_0]d\tau, \tag{4.17}$$

for RHAM2

$$u_1^{RHAM2} = \frac{\hbar}{(1+t)}\int_0^t \frac{1}{(1+\tau)^2}R_1[\vec{u}_0]d\tau = -\hbar\frac{t}{(1+t)^2}\wp(u_0), \tag{4.18}$$

$$u_2^{RHAM2} = u_1^{RHAM2} + \frac{\hbar}{1+t}\int_0^t \frac{1}{(1+\tau)^2}R_2[u_1^{RHAM2}, u_0]d\tau. \tag{4.19}$$

Both terms u_2^{RHAM1} and u_2^{RHAM2} can be easily found in the explicit form. In what follows, we give some more details about RHAM2 and in the same way RHAM1 can be calculated. The first term obtained, u_1^{RHAM2}, is a composition of two functions, one depending on the variable t and the other one on $\mathbf{x}$, so we can write $u_1^{RHAM2} = -\hbar\frac{t}{(1+t)^2}\wp(u_0) = f(t)w(\mathbf{x})$, with $f(t) = -\hbar\frac{t}{(1+t)^2}$ and $w(\mathbf{x}) = \wp(u_0)$. In this way the

$R_2[u_1, u_0]$ term can be rewritten

$$\begin{aligned} R_2[u_1, u_0] &= f_t(t)w(\mathbf{x}) - f(t)\frac{\partial u}{\partial x}\left(\frac{w_x}{|\nabla u_0|_\beta} - \frac{u_{0x}w_x + u_{0y}w_y}{|\nabla u_0|_\beta^3}u_{0x}\right) \\ &\quad - f(t)\frac{\partial u}{\partial y}\left(\frac{w_y}{|\nabla u_0|_\beta} - \frac{u_{0x}w_x + u_{0y}w_y}{|\nabla u_0|_\beta^3}u_{0y}\right) + \lambda f(t)w(\mathbf{x}) \\ R_2[u_1, u_0] &= f_t(t)w(\mathbf{x}) - f(t)\left\{S(w(\mathbf{x}), w_x(\mathbf{x}), w_y(\mathbf{x}), u_0(\mathbf{x}))\right\} \end{aligned}$$

where

$$S(w, w_x, w_y, u_0) = \frac{\partial}{\partial x}\left(\frac{w_x}{|\nabla u_0|_\beta} - \frac{u_{0x}w_x + u_{0y}w_y}{|\nabla u_0|_\beta^3}u_{0x}\right) + \frac{\partial}{\partial y}\left(\frac{w_y}{|\nabla u_0|_\beta} - \frac{u_{0x}w_x + u_{0y}w_y}{|\nabla u_0|_\beta^3}u_{0y}\right) + \lambda w$$

and from (4.19), we have

$$\begin{aligned} u_2^{RHAM2} &= f(t)w(\mathbf{x}) + \frac{\hbar}{1+t}\left[w(\mathbf{x})\int_0^t \frac{1}{(1+\tau)^2}f_t(\tau)d\tau - \right. \\ &\quad \left. -S(w(\mathbf{x}), w_x(\mathbf{x}), w_y(\mathbf{x}), u_0(\mathbf{x}))\int_0^t \frac{1}{(1+\tau)^2}f(t)d\tau\right] \end{aligned}$$

$$u_2^{RHAM2} = -\hbar\frac{t}{(1+t)^2}w + \frac{\hbar}{1+t}\left[w\hbar\frac{1-2t}{6(1+t)^4} - S(w, w_x, w_y, u_0)\hbar\frac{1+3t}{6(1+t)^3}\right].$$

An immediate observation is that the computational cost of the first order approximation of the RHAM is as expensive as the TM. The second order term of RHAM, u_2, brings the extra computation of the term $S(w, w_x, w_y, u_0)$ which is obtained explicitly without having too much extra computation since the curvature term $w(x, y)$ has been already evaluated in the same step. The numerical implementation for RHAM1 from equations (4.16) and (4.17) can be given in the form of an algorithm as described in Algorithm 3. In the same way considering (4.18) and (4.19) we get Algorithm 4 which describes RHAM2.

Algorithm 3 Restarted Homotopy Analysis Method solution expressed by exponential functions RHAM1: $u^k \leftarrow RHAM1(u^k, u_0, \lambda, \beta, \theta, \Delta t, maxit, tol)$.

Initialization: Given $u^0_{i,j} = u_{i,j,0} = u_{0,i,j}$,
for $k = 1 : maxit$ **do**
Set $u^k_{i,j,0} := u^{k-1}_{i,j}$
Compute $\wp(u^k_{i,j,0})$ using (4.9),
Compute $u^k_{i,j,1}$ using (4.16),
Compute $u^k_{i,j,2}$ using (4.17),
$u^k_{i,j} \leftarrow u^k_{i,j,0} + u^k_{i,j,1} + u^k_{i,j,2}$
If $\|u^k - u^{k-1}\| < tol$ or $PSNR(u^{k-1}) > PSNR(u^k)$, set $u^k_{i,j} \leftarrow u^{k-1}_{i,j}$, Break;
$u^k_{i,j} \leftarrow u^{k-1}_{i,j}$

end for

Algorithm 4 Restarted Homotopy Analysis Method solution expressed by fractional functions RHAM2: $u^k \leftarrow RHAM2(u^k, u_0, \lambda, \beta, \Delta t, maxit, tol)$.

Initialization: Given $u^0_{i,j} = u_{i,j,0} = u_{o,i,j}$,
for $k = 1 : maxit$ **do**
Set $u^k_{i,j,0} := u^{k-1}_{i,j}$
Compute $\wp(u^k_{i,j,0})$ using (4.9),
Compute $u^k_{i,j,1}$ using (4.18),
Compute $u^k_{i,j2}$ using (4.19),
$u^k_{i,j} \leftarrow u^k_{i,j,0} + u^k_{i,j,1} + u^k_{i,j,2}$
If $\|u^k - u^{k-1}\| < tol$ or $PSNR(u^{k-1}) > PSNR(u^k)$, set $u^k_{i,j} \leftarrow u^{k-1}_{i,j}$, Break;
$u^k_{i,j} \leftarrow u^{k-1}_{i,j}$

end for

Results in Section §4.5 will show that the RHAMs are effective for solving nonlinear equations such as equation (4.13). There are many models using the TV operators as in (4.13) which might be solved by RHAM. Below we give one such application of RHAM to a TV variational model in image segmentation known for its better performance in the presence of noise.

4.4 Application of RHAM to a Variational Image Segmentation Model in $\mathbb{R}^2$ and $\mathbb{R}^3$

Recall from Section §3.4 that given an image $u_0(\mathbf{x}) : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ in a bounded domain Ω, segmentation refers to separation into two subregions, features and their background, with internal boundary Γ in Ω. For $n = 3$, Γ is a 2-D surface separating the two 3-D volumetric regions $\Omega \subset \mathbb{R}^3$ into inside(Γ) and outside(Γ) such that $\Omega = inside(\Gamma) \cup \Gamma \cup outside(\Gamma)$. As detailed in §3.4 the Chan-Vese method [37] considers the image as piecewise continuous functions that are surrounded by discontinuities

represented as contours/surfaces. The Euler-Lagrange equation for the model is written as follows:

$$\begin{cases} \delta_\epsilon(\phi(\mathbf{x}))\Big[\mu\nabla\cdot\Big(\dfrac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\Big) - f_0(\phi(\mathbf{x}))\Big] = 0 & \text{in } \Omega, \\ \dfrac{\delta_\epsilon(\phi(\mathbf{x}))}{|\nabla\phi(\mathbf{x})|}\dfrac{\partial\phi(\mathbf{x})}{\partial\vec{n}} = 0 & \text{on } \partial\Omega, \end{cases} \tag{4.20}$$

where $\vec{n}$ is the unit normal exterior to the boundary $\partial\Omega$, $\dfrac{\partial\phi}{\partial\vec{n}}$ is the normal derivative of ϕ at the boundary and

$$f_0(\phi(\mathbf{x})) = \lambda_1(u_0(\mathbf{x}) - c_1(\phi(\mathbf{x})))^2 - \lambda_2(u_0)(\mathbf{x}) - c_2(\phi(\mathbf{x})))^2 \tag{4.21}$$

with

$$c_1(\phi(\mathbf{x})) = \frac{\int_\Omega u_0(\mathbf{x})H_\epsilon(\phi(\mathbf{x}))d\mathbf{x}}{\int_\Omega H_\epsilon(\phi(\mathbf{x}))d\mathbf{x}} \tag{4.22}$$

$$c_2(\phi(\mathbf{x})) = \frac{\int_\Omega u_0(\mathbf{x})(1 - H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}}{\int_\Omega(1 - H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}}, \tag{4.23}$$

(i.e the curve has a nonempty interior and exterior in Ω). We remark that previously in Chapter 3 and also in Chan-Vese paper [37], we treated c_1 and c_2 explicitly in alternating iteration fashion but here we consider $f_0(\phi(\mathbf{x}))$ fully coupled with ϕ for the following derivation of RHAM.

In a similar way to equation (4.13) we can consider choosing the nonlinear operator

$$\mathcal{N}[\varphi)] = \frac{\partial\varphi}{\partial t} - \delta_\epsilon(\varphi)\Big[\mu\nabla\cdot\Big(\frac{\nabla\varphi}{|\nabla\varphi|}\Big) - f_0(\varphi)\Big], \tag{4.24}$$

where $0 \le q \le 1$, $\varphi = \varphi(\mathbf{x}, t; q)$ and with the initial approximation for the level set $\phi(\mathbf{x}, \mathbf{0})$ a circle/sphere with given center and radius $\mathbf{x}_0$ and r, respectively. The role of u_0, u_1, u_2 in the previous section is carried out by ϕ_0, ϕ_1 ϕ_2. Instead of updating ϕ from ϕ_0 as in TM, we now decompose ϕ as $\phi = \phi_0 + \phi_1 + \phi_2$ by getting in this way a second order approximation before restarting. In this way we can update $\phi = \phi_0 + \phi_1 + \phi_2$ after we apply $\mathcal{L}^{-1}$ in both cases.

We will show the 3-D discretization realization for the segmentation case due to its importance in 3-D applications and as a more general case compared to 2-D. Let us assume that the domain $\Omega \in \mathbb{R}^3$ has been split into $N \times M \times L$ cells where the grid points are located at $(x_i = ih_x, y_j = jh_y, z_l = lh_z)$, $i = 1, ..., N$, $j = 1, ..., M$, $l = 1, ..., L$, defining a real-coordinate system bounded by the image domain $\Omega \in \mathbb{R}^3$, and $t_k = k\Delta t$, where Δt and $k = 1, 2, ...$, denotes the time step and iteration number, respectively. We denote the values of $\phi(x, y, z, t)$ at the grid points (x_i, y_j, z_l, t_k) by $\phi^k_{i,j,l}$, and $u^0_{i,j,l} = u_0(x_i, y_j, z_l)$. For simplicity and without lost of generality we assume that $M = N = L$, $h_x = h_y = h_z = 1$. Using finite differences with equation (4.20) we obtain the time marching method in the following form

$$\phi^{k+1}_{i,j,l} = \phi^k_{i,j,l} + \Delta t\delta_{i,j,k}\wp(\phi^k_{i,j,l}), \tag{4.25}$$

where $\wp(\phi^k_{i,j,l}) = \mu\kappa(\phi^k_{i,j,l}) - f_0{}^k_{i,j,l}$ and $\kappa(\phi^k_{i,j,l})$ is the 3-D curvature calculated in the following form

$$\begin{array}{c} \kappa(\phi^k_{i,j,l}) = \nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|_\beta}\right)\Big|_{i,j,l} = \left(\frac{\partial}{\partial x}\left[\frac{\phi_x}{|\nabla\phi_\beta|}\right] + \frac{\partial}{\partial y}\left[\frac{\phi_x}{|\nabla\phi_\beta|}\right] + \right. \\ \left.\frac{\partial}{\partial z}\left[\frac{\phi_z}{|\nabla\phi_\beta|}\right]\right)\Big|_{i,j,l} = \Delta^x_-\left(\frac{\Delta^x_+\phi^k_{ij}}{\sqrt{(\Delta^x_+\phi^k_{ij})^2+(\Delta^y_+\phi^k_{ij})^2+\beta}}\right) \\ +\Delta^y_-\left(\frac{\Delta^y_+\phi^k_{ij}}{\sqrt{(\Delta^y_+\phi^k_{ij})^2+(\Delta^y_+\phi^k_{ij})^2+\beta}}\right) + \Delta^z_-\left(\frac{\Delta^z_+\phi^k_{ij}}{\sqrt{(\Delta^z_+\phi^k_{ij})^2+(\Delta^z_+\phi^k_{ij})^2+\beta}}\right) \end{array} ,$$

with $\Delta^x_\mp\phi_{i,j,l}, \Delta^y_\mp\phi_{i,j,l}, \Delta^z_\mp\phi_{i,j,l}$ first order finite differences.

For the linear operator $\mathcal{L} = \mathcal{L}_1$ we get:

$$\begin{array}{rcl} \phi_1 = & \hbar exp(-\theta t)\int_0^t exp((\theta-2)\tau)R_1[\vec{\phi}_0]d\tau & \\ = & -\hbar\frac{e^{-2t}-e^{-\theta t}}{\theta-2}\delta_\epsilon(\phi_0)\Big[\mu\nabla\cdot\Big(\frac{\nabla\phi_0}{|\nabla\phi_0|}\Big) - f_0(\phi_0)\Big], & \end{array} \tag{4.26}$$

$$\phi_2 = \phi_1 + \hbar exp(-\theta t)\int_0^t exp((\theta-2)\tau)R_2[\phi_0,\phi_1]d\tau, \tag{4.27}$$

which can be easily written in the explicit way by considering $R_1[\vec{\phi}_0]$ and $R_2[\vec{\phi}_0]$ given in (4.30) and (4.31). In the same way applying $\mathcal{L}^{-1} = \mathcal{L}_2^{-1}$ we get

$$\begin{array}{rcl} \phi_1 = & \frac{\hbar}{(1+t)}\int\limits_0^t \frac{1}{(1+\tau)^2}R_1[\vec{u}_0]d\tau & \\ = & -\hbar\frac{t}{(1+t)^2}\delta_\epsilon(\phi_0)\Big[\mu\nabla\cdot\Big(\frac{\nabla\phi_0}{|\nabla\phi_0|}\Big) - f_0(\phi_0)\Big], & \end{array} \tag{4.28}$$

$$\phi_2 = \phi_1 + \frac{\hbar}{1+t}\int_0^t \frac{1}{(1+\tau)^2}R_2[\phi_0,\phi_1]d\tau. \tag{4.29}$$

In the same way we derive the 3-D expression for $R_1[\vec{\phi}_0]$ and $R_2[\vec{\phi}_0]$ from equation (4.3) as follows:

$$\begin{array}{c} R_1[\phi_0] = \frac{\partial^0 N[\varphi(\mathbf{x},t;q)]}{\partial^0 q}\Big|_{q=0} = \mathcal{N}(\phi_0) = \\ -\delta_\epsilon(\phi_0)\Big[\mu\nabla\cdot\Big(\frac{\nabla\phi_0}{|\nabla\phi_0|}\Big) - f_0(\phi_0)\Big], \end{array} \tag{4.30}$$

$$
\begin{aligned}
R_2[\phi_0,\phi_1] &= \frac{\partial N[\varphi(\mathbf{x},t;q)]}{\partial q}\Big|_{q=0} \\
&= \frac{\partial}{\partial q}\Big(\frac{\partial \varphi}{\partial t} - \delta_\epsilon(\varphi)\Big[\mu\nabla\cdot\Big(\frac{\nabla\varphi}{|\nabla\varphi|}\Big) - f_0(\varphi)\Big]\Big)\Big|_{q=0} \\
&= \Big(\frac{\partial}{\partial q}\frac{\partial \varphi}{\partial t} - \frac{\partial}{\partial q}\Big(\delta_\epsilon(\varphi)\Big)\Big[\mu\nabla\cdot\Big(\frac{\nabla\varphi}{|\nabla\varphi|}\Big) - f_0(\varphi)\Big] - \\
&- \delta_\epsilon(\varphi)\frac{\partial}{\partial q}\Big[\mu\nabla\cdot\Big(\frac{\nabla\varphi}{|\nabla\varphi|}\Big) - f_0(\varphi)\Big]\Big)\Big|_{q=0} \\
&= (\frac{\partial}{\partial t}\frac{\partial \varphi}{\partial q} - \frac{\partial \varphi}{\partial q}\delta'_\epsilon(\varphi)\Big[\mu\nabla\cdot\Big(\frac{\nabla\varphi}{|\nabla\varphi|}\Big) - f_0(\varphi)\Big] - \\
&- \delta_\epsilon(\varphi)\Big[\mu\frac{\partial}{\partial q}\nabla\cdot\Big(\frac{\nabla\varphi}{|\nabla\varphi|}\Big) - \frac{\partial f_0(\varphi)}{\partial q}\Big])\Big|_{q=0} \\
&= \phi_{1t} - \phi_1\delta'_\epsilon(\phi_0)\Big[\mu\nabla\cdot\Big(\frac{\nabla\phi_0}{|\nabla\phi_0|}\Big) - f_0(\phi_0)\Big] \\
&- \delta_\epsilon(\phi_0)\Big[\mu\sum_{i=1}^{n}\frac{\partial}{\partial x_i}(\frac{\phi_{1i}}{|\nabla\phi_0|_\beta} - \frac{\sum_{i=1}^{n}\phi_{0x_i}\phi_{1x_i}}{|\nabla\phi_0|_\beta^3}\phi_{0x_i}) - f'_0(\phi_0)\Big],
\end{aligned}
\tag{4.31}
$$

where $n = 3$ is the space dimension,

$$
\delta'_\epsilon(\phi_0) = \frac{\varepsilon}{\pi}\frac{-2\phi_0}{(\varepsilon^2+\phi_0^2)^2} \tag{4.32}
$$

and

$$
f'_0(\phi_0) = -2c'_1(\phi_0)\lambda_1(u_0 - c_1(\phi_0)) + 2c'_2(\phi_0)\lambda_2(u_0 - c_2(\phi_0)).
$$

To obtain $f'_0(\phi_0)$, we also need to calculate[1]

$$
c'_1(\phi_0) = \frac{w_1 - w_2}{(\int_\Omega H_\epsilon(\phi_0)d\mathbf{x})^2}, c'_2(\phi_0) = \frac{-w_3 + w_4}{(\int_\Omega(1 - H_\epsilon(\phi_0))d\mathbf{x})^2},
$$

$$
\begin{aligned}
w_1 &= (\int_\Omega u_0\delta_\epsilon(\phi_0)\phi_1 d\mathbf{x})(\int_\Omega H_\epsilon(\phi_0)d\mathbf{x}), \\
w_2 &= (\int_\Omega \delta_\epsilon(\phi_0)\phi_1 d\mathbf{x})(\int_\Omega u_0 H_\epsilon(\phi_0)d\mathbf{x}).
\end{aligned}
$$

From the above equation we can notice that time marching would be a simple RHAM, for $m = 1$ and without any auxiliary function. The algorithms for the 3-D discrete case are described in Algorithms 5 and 6.

[1] to calculate c'_1 and c'_2 in $f'_0(\phi_0)$ the chain rule differentiation formula has been employed and the differentiation is with respect to q.

Algorithm 5 3-D Iterative Homotopy Analysis Method Solution Expressed by Exponential Functions RHAM1: $\phi^k \leftarrow RHAM1(\phi^k, u_0, \lambda, \beta, \theta, \Delta t, maxit, tol)$.

Initialization: Given $\phi^0_{i,j,l} = \phi_{i,j,l,0}$,
for $k = 1 : maxit$ **do**
  Set $\phi^k_{i,j,l,0} := \phi^{k-1}_{i,j,l}$,
  Compute $\phi^k_{i,j,l,1}$ using (4.26),
  Compute $\phi^k_{i,j,l,2}$ using (4.27),
  $\phi^k_{i,j,l} \leftarrow \phi^k_{i,j,l,0} + \phi^k_{i,j,l,1} + \phi^k_{i,j,l,2}$
  If $\|\phi^k - \phi^{k-1}\| < tol$ set $\phi^k_{i,j,l} \leftarrow \phi^{k-1}_{i,j,l}$, Break;
  $\phi^k_{i,j,l} \leftarrow \phi^{k-1}_{i,j,l}$,

end for

Algorithm 6 3-D Iterative Homotopy Analysis Method Solution Expressed by Fractional Functions RHAM2: $\phi^k \leftarrow RHAM2(\phi^k, u_0, \lambda, \beta, \Delta t, maxit, tol)$.

Initialization: Given $\phi^0_{i,j,l} = \phi_{i,j,l,0}$,
for $k = 1 : maxit$ **do**
  Set $\phi^k_{i,j,l,0} := \phi^{k-1}_{i,j,l}$
  Compute $\phi^k_{i,j,l,1}$ using (4.28)
  Compute $\phi^k_{i,j,l,2}$ using (4.29)
  $\phi^k_{i,j,l} \leftarrow \phi^k_{i,j,l,0} + \phi^k_{i,j,l,1} + \phi^k_{i,j,l,2}$
  If $\|\phi^k - \phi^{k-1}\| < tol$ set $\phi^k_{i,j,l} \leftarrow \phi^{k-1}_{i,j,l}$, Break;
  $\phi^k_{i,j,l} \leftarrow \phi^{k-1}_{i,j,l}$,

end for

4.5 Experimental Results for the Denoising and Image Segmentation Problems Using the RHAM

In this section, we shall illustrate the speed and accuracy of RHAM, providing experimental results in denoising and segmentation for different artificial, synthetic and real images. All the 2-D experiments has been processed using a Pentium(R) Dual-Core T4200 2.00GHz CPU, 2.92 GB of RAM computer while the 3-D experiments on a Intel(R) Core TM i7-2600 3.40GHz CPU, 16 GB of RAM computer. In the experimental section we will compare our method not only with time marching but with a fast solver as well, such as the AOS method [174] which is an alternating direction implicit method [129].

To measure the *restoration quality* the parameters **SNR** and **PSNR** can be used. **SNR** (the signal to noise ratio) of a noisy image is a measure of how much noise is presented in the image, and the **PSNR** (the peak signal to noise ratio) measures how

close the images are to each-other. These quantities are given by the following formulas:

$$SNR = 10\log_{10}\Big(\frac{\sum_{(i,j)}(u_{i,j})^2}{\sum_{(i,j)}(u_{i,j}-v_{i,j})^2}\Big) \text{ and } PSNR = 10\log_{10}\frac{255^2}{RMSE}$$
$$\text{with} \quad RMSE = \sqrt{\frac{1}{n_1 n_2}\sum_{(i,j)}(u_{i,j}-v_{i,j})^2} \tag{4.33}$$

where $u_{i,j}$ is the value of the true image at the grid point (i,j), $v_{i,j}$ is the value of the restored image at the grid point (i,j), $n_1 n_2$ is the total number of pixels. The larger the **SNR (PSNR)** is, the better the restored image is. In real life situations, such a measure is not possible because u is not known. In real applications, using **PSNR** was found to be more useful.

In the case of image segmentation the relative residual has been used as an indicator of reaching the boundaries.

4.5.1 Test Set 1 – results for the denoising problem using the RHAM method

In this subsection we test our algorithms on denoising in several images with an intensity range $[0, 255]$ which includes real life and artificial images. The image size shown is 256×256 and the parameter values $\beta = 10^{-6}$, $\Delta t = 0.01$ and $maxit = 1000$ have been fixed. Random normal noise of $10\%, 15\%$, and 20% has been used with λ varying at $0.1, 0.05$, and 0.01, respectively. Figures 4.3, 4.4, 4.5 and 4.6 show the results obtained with the two new methods described in Algorithms 3 and 4 above in comparison with the time marching method for Gauss white noise. With the same parameter λ the experiments has been repeated for uniform noise of $10\%, 30\%$, and 50%. The experiments show that the model works for both these type of noise. Tables 4.1 and 4.2 show the comparison in terms of CPU time and number of iterations required for those methods in the case of random noise while Tables 4.3 and 4.4 show the results in the case of uniform random noise. In table 4.5 we show the results obtained from applying the AOS method for the same figures in the case of random normal noise with different Δt. From tables 4.1, 4.2, 4.3 and 4.4 we can notice a great speed increase with the RHAMs in comparison with time marching, while table 4.5 shows that the AOS method is slow for small Δt and loses some accuracy if we increase Δt to 1. The bold numbers in the PSNR columns in table 4.5 show that the PSNR of both AOS and the RHAMs have been the same, the non-bold PSNRs show the maximum that the AOS method could achieve. In the experiments it has been noticed that when the level of noise increases, increasing the parameter θ for RHAM1 gives better results. For this reason in cases of high noise, we set $\theta = 1$ instead of $\theta = 0.01$ considered for low noise in RHAM1.

Figure 4.3: Test Set-1 Example 1. From the top row to the bottom, 10, 15 and 20 percent random noise has been added to the image. For each row the first image is the noisy image, the second shows the result obtained with the TM method for the same number of iterations as the maximum iterations used for RHAM1 and RHAM2, the third and the forth images show the results obtained for RHAM1 and RHAM2.

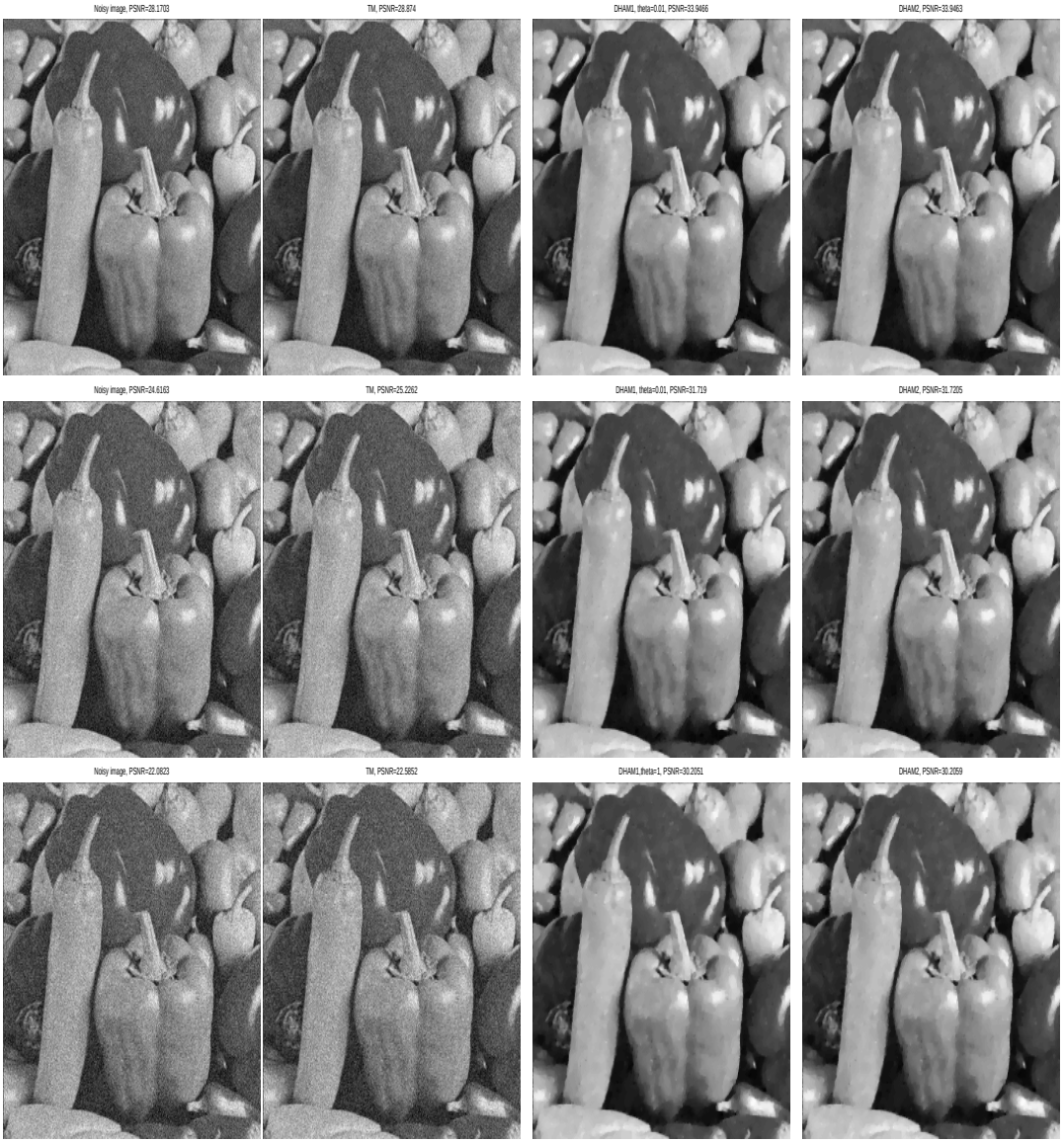


Figure 4.4: Test Set-1 Example 2. From the top row to the bottom, 10, 15 and 20 percent random noise has been added to the image. For each row the first image is the noisy image, the second shows the result obtained with the TM method for the same number of iterations as the maximum iterations used for RHAM1 and RHAM2, the third and the forth images show the results obtained for RHAM1 and RHAM2.

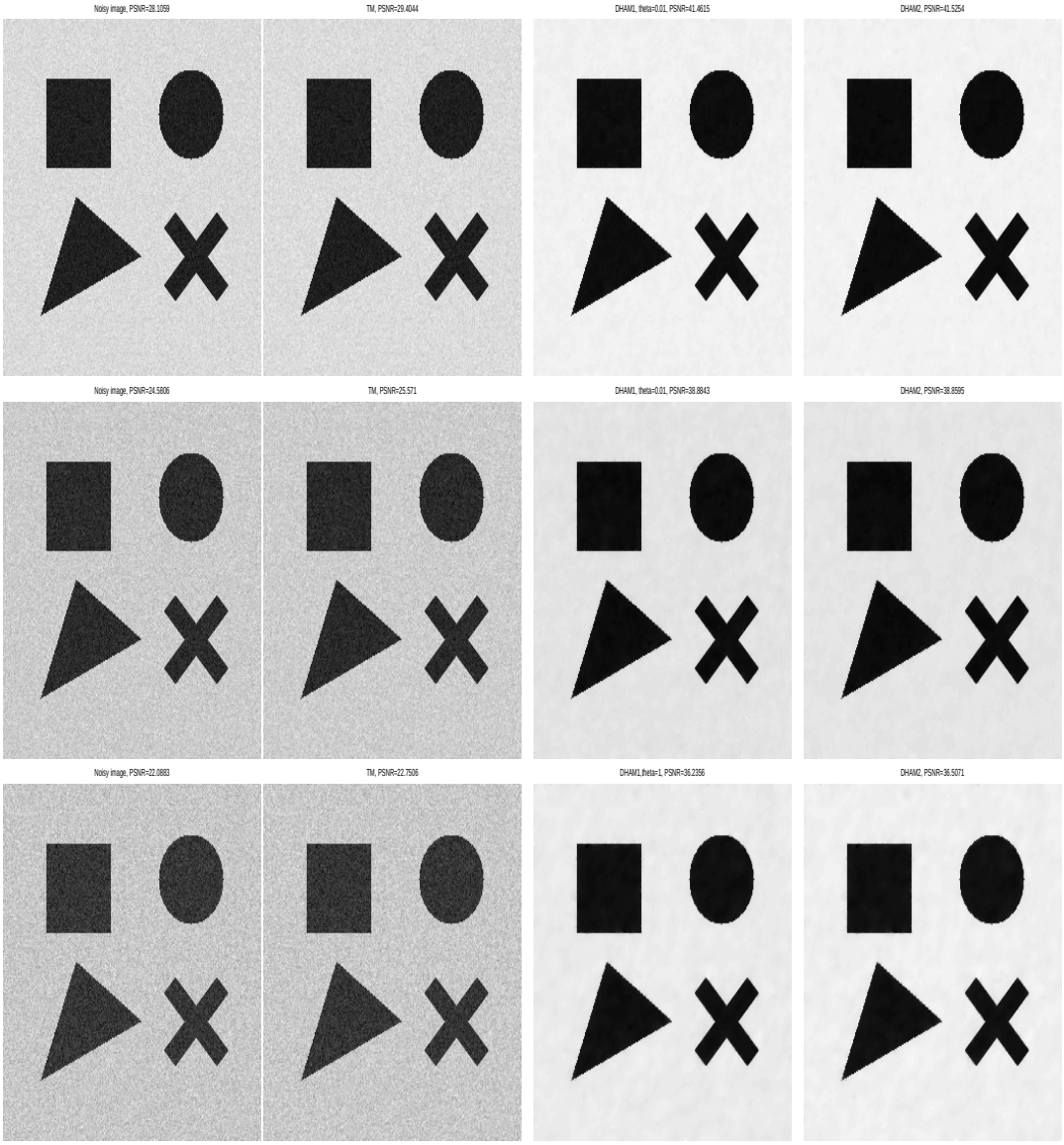


Figure 4.5: Test Set-1 Example 3. From the top row to the bottom, 10, 15 and 20 percent random noise has been added to the image. For each row the first image is the noisy image, the second shows the result obtained with the TM method for the same number of iterations with maximum of iteration used for RHAM1 and RHAM2, the third and the forth images show the results obtained for RHAM1 and RHAM2.

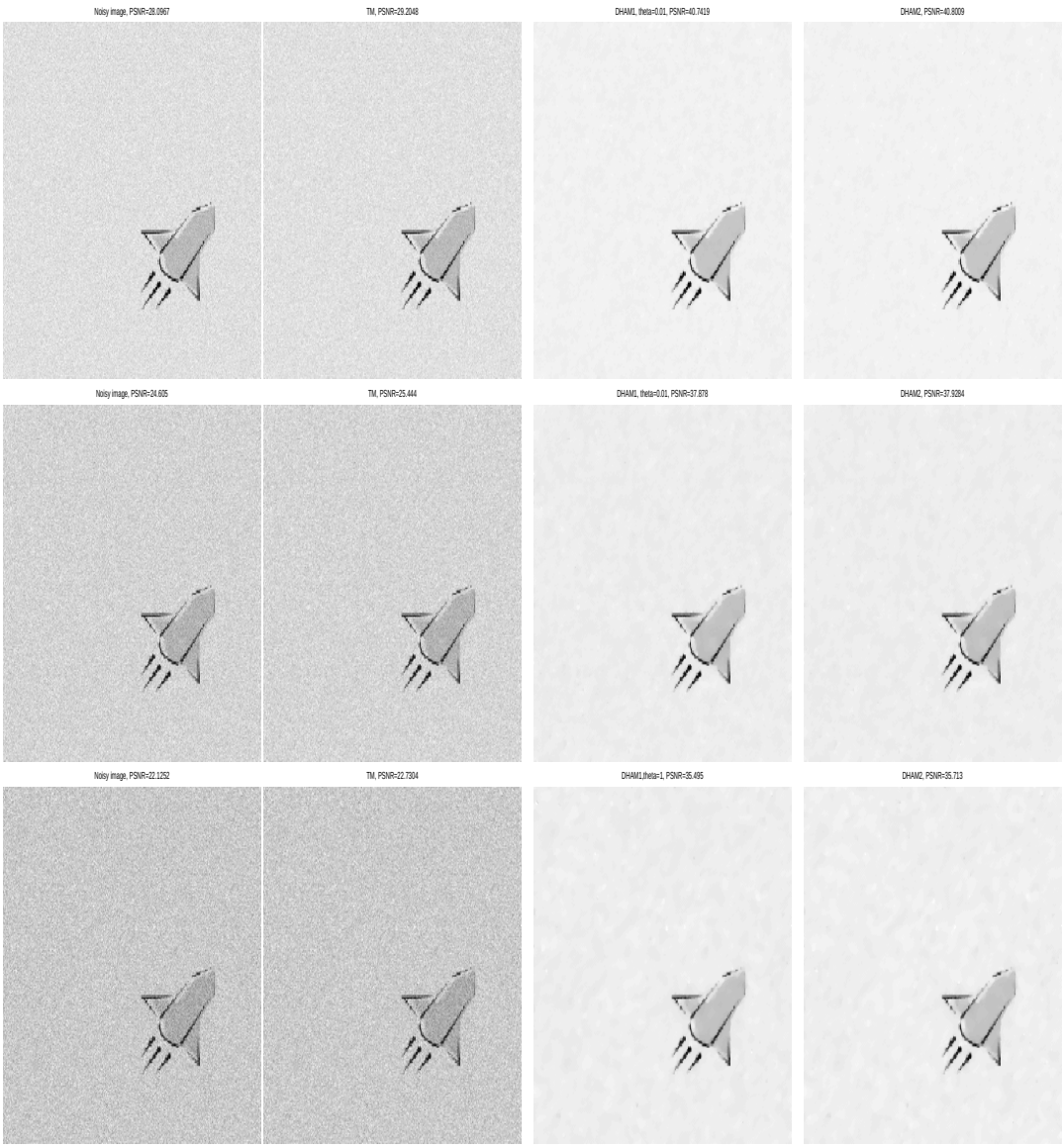
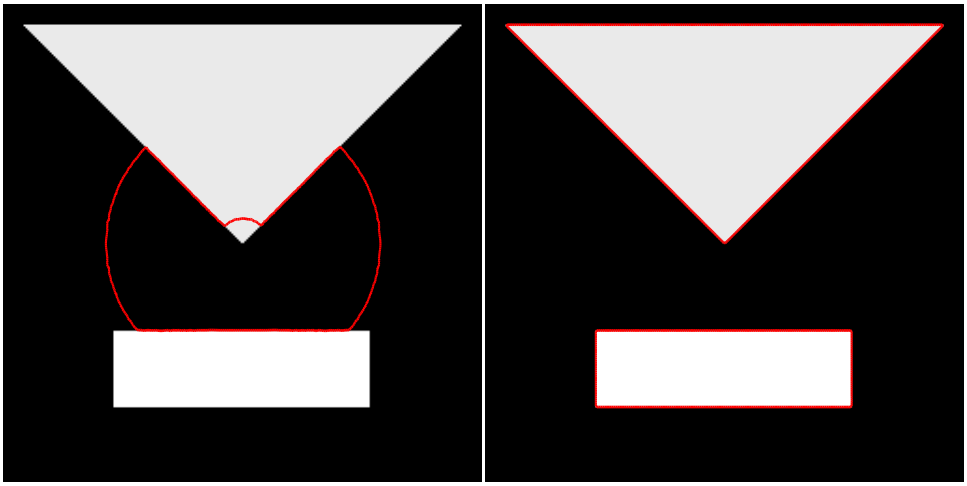


Figure 4.6: Test Set-1 Example 4. From the top row to the bottom, 10, 15 and 20 percent random noise has been added to the image. For each row the first image is the noisy image, the second shows the result obtained with the TM method for the same number of iterations as the maximum iterations used for RHAM1 and RHAM2, the third and the forth images show the results obtained for RHAM1 and RHAM2.

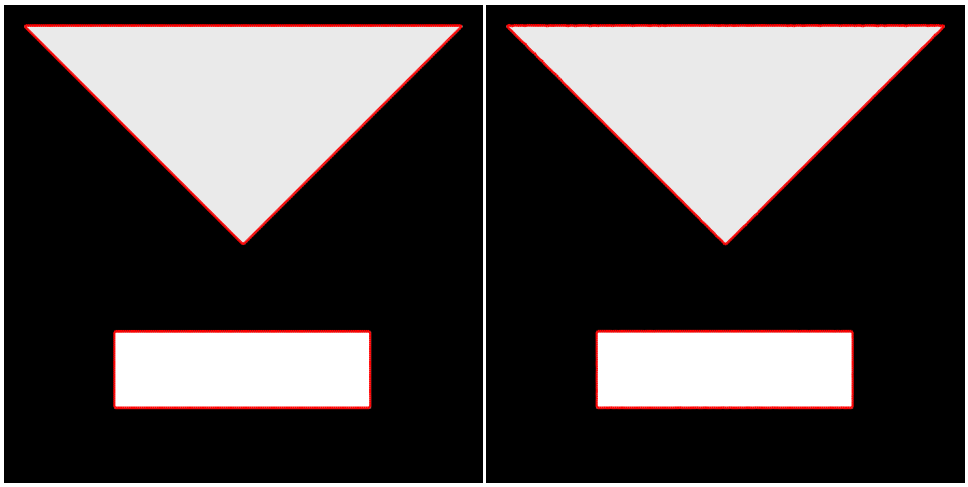
4.5.2 Test Set 2 – results for segmentation using RHAM method

In the following experiments the parameters θ, Δt, μ, λ_1, λ_2, h (step size) and $\hbar$ have been fixed as follows: $\theta = 1$, $\Delta t = 0.01$, $\mu = 1$, $\lambda_1 = \lambda_2 = 50$, $h = 1$ and $\hbar = -1$. The size of the image considered in the following examples is $n = 256$ and the initial level set is placed in the center of the image and the radius is 50. To stop the program, the relative residual 10^{-2} or 10^{-3} has been used. Table 4.9 shows the comparison between the TM method and RHAM1 and RHAM2 in CPU time and maximum iterations needed to have the given residual as shown in each of the pictures. It can be easily noticed that both RHAM1 and RHAM2 are at least ten times faster than the TM method. It is well known that for the AOS method Δt can be chosen sufficiently



(a) Time marching method result after 32 iterations

(b) HAM1 method result after 32 iterations

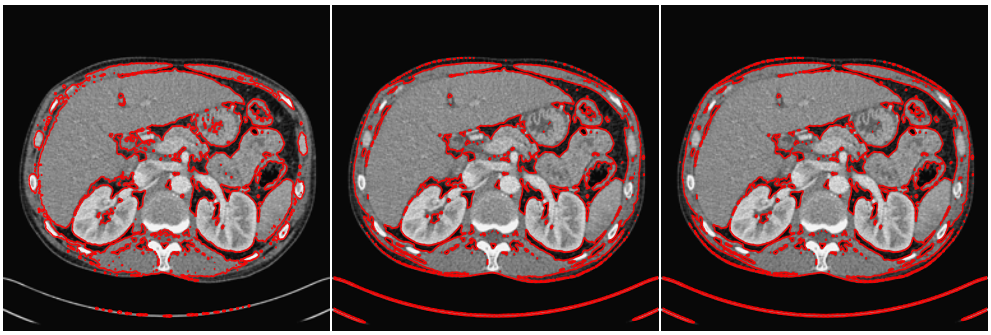


(c) HAM2 method result after 32 iterations

(d) Time marching method result after 1287 iterations

Figure 4.7: Test Set-2 Example 1. Successfully reached relative residual equal to 10^{-2} after 32 iterations for the RHAM's and 1287 iterations for TM.

large that we get fast convergence. In the following experiments we show the results obtained comparing the RHAM1 with the AOS method. Figures 4.12 and 4.13 show the



(a) Time marching method result after 33 iterations

(b) HAM1 method result after 33 iterations

(c) HAM2 method result after 33 iterations

Figure 4.8: Test Set-2 Example 2. Successfully reached relative residual equal to 10^{-3} after 33 iterations for the RHAMs and 205 iterations for TM.

(a) Time marching method result after 9 iterations

(b) HAM1 method result after 9 iterations

(c) HAM2 method result after 9 iterations

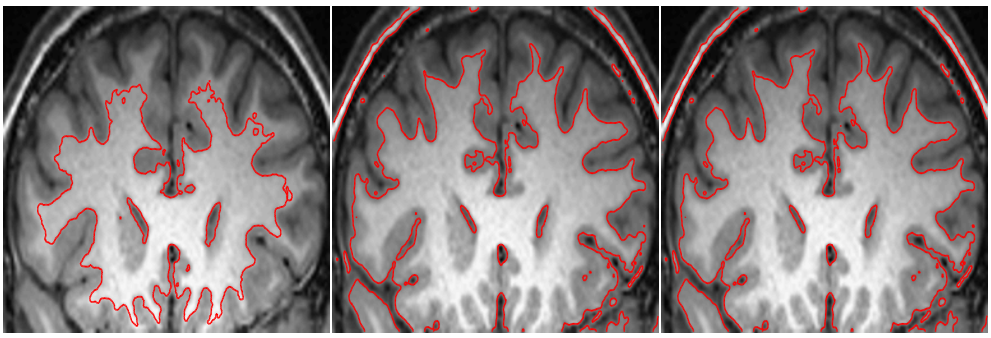
Figure 4.9: Test Set-2 Example 3. Successfully reached relative residual equal to 10^{-2} after 9 iterations for the RHAMs and 186 iterations for TM.

(a) Time marching method result after 53 iterations

(b) HAM1 method result after 53 iterations

(c) HAM2 method result after 53 iterations

Figure 4.10: Test Set-2 Example 4. Successfully reached relative residual equal to 10^{-3} after 53 iterations for the RHAMs and 929 iterations for TM.



(a) Time marching method result after 47 iterations (b) HAM1 method result after 47 iterations (c) HAM2 method result after 47 iterations

Figure 4.11: Test Set-2 Example 5. Successfully reached relative residual equal to 10^{-3} after 47 iterations for RHAMs and 467 iterations for TM.

results obtained for $\Delta t = 0.01$ while Tables 4.7 and 4.8 show the comparison between the methods for different values of Δt. It can be noticed that the RHAM method is at least two times faster than the AOS method.

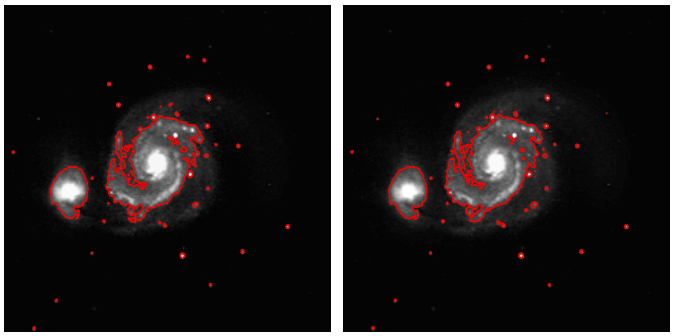


Figure 4.12: Test Set-2 Example 6. Successfully segmented after 59 iteration for RHAM2 and 1558 for AOS method, reaching relative residual equal to 10^{-3} .

4.5.3 Test Set 3 – results for three-dimensional image segmentation problem using the RHAM method

The speed of the solution is a particular issue for three dimensional data. In this subsection, in order to illustrate the speed and the accuracy of the 3-D RHAM segmentation algorithms we evaluate the performance of the proposed method on artificial, CT and MRI 3-D data sets in terms of segmentation quality and computational speed. The relative residual has been used as an indicator of reaching the boundaries. In the following experiments the parameters θ, Δt, μ, λ_1, λ_2, h (step size), ϵ and $\hbar$ have been fixed at $\theta = 0.01$, $\Delta t = 0.01$, $\mu = 1$, $\lambda_1 = \lambda_2 = 50$, $\epsilon = 1$ and $\hbar = -1$ respectively. The initial level set is placed in the center of the image with radius 40 and to stop the program, the relative residuals 10^{-1}, 10^{-2}, 10^{-3} or 10^{-4} has been used.

Figures 4.14, 4.15(a), 4.15(b) and 4.15(c) show the results obtained for 3-D with the two new methods described in comparison with the time marching method for artificial images, while Figure 4.16 displays real 3-D CT data. The results show that

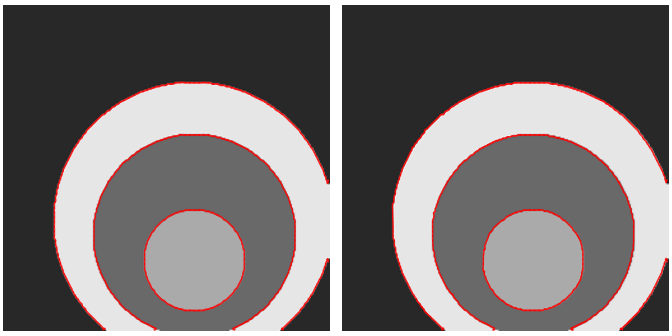


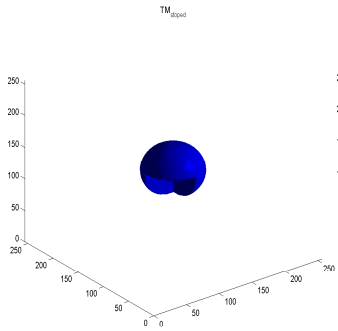
Figure 4.13: Test Set-2 Example 7. Successfully segmented after 31 iteration for RHAM2 and 496 for the AOS method, reaching relative residual equal to 10^{-3} .

the proposed approach improves the speed of the segmentation significantly. Table 4.9 compares the segmentation performance of our RHAM to the Chan-Vese active contour model [37] in CPU time and maximum iteration needed to have the given residual as shown in each of the pictures. It can be easily noticed that both RHAM1 and RHAM2 are at least ten times faster than the TM method.

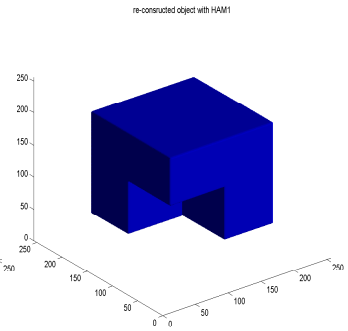
More experiments has been carried out with CT data as shown in Figures 4.17, 4.18, 4.19. It can be noticed that the RHAM has the same segmentation accuracy as the TM method but in less time.

4.6 Conclusion

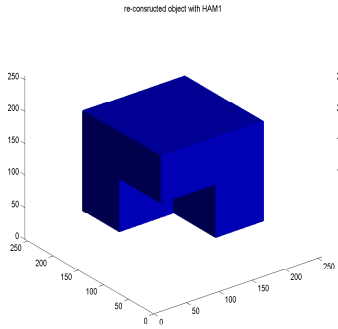
In this chapter, we proposed a discrete restarted homotopy analysis method to obtain numerical solutions for image denoising and segmentation which are important and practical problems in image processing. The experimental results show that the method is effective in giving fast solutions for denoising and segmentation, particularly crucial in 3-D. On the other hand the method is reliable and with higher accuracy compared with other methods. Numerical tests demonstrate that RHAM shows great speed compared with time marching and is faster than the AOS method. Our results in denoising and segmentation suggest that RHAM may be used to potentially solve other nonlinear PDEs arising from image processing models, which will be our future work.



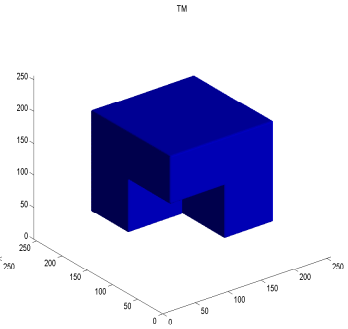
(a) Time marching method result after 2 iterations



(b) RHAM1 method result after 2 iterations

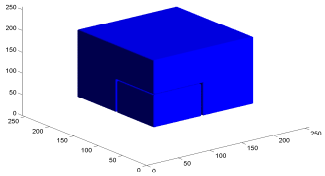


(c) RHAM2 method result after 2 iterations

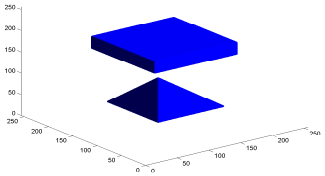


(d) Time marching method result after 1000 iterations

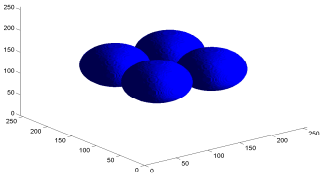
Figure 4.14: Test Set-3 Example 1. Successfully reached relative residual equal to 10^{-4} after 2 iterations for the RHAMs and 1000 iterations for TM.



(a) Successfully reached relative residual equal to 10^{-4} after 2 iterations for the RHAMs and 800 iterations for TM.

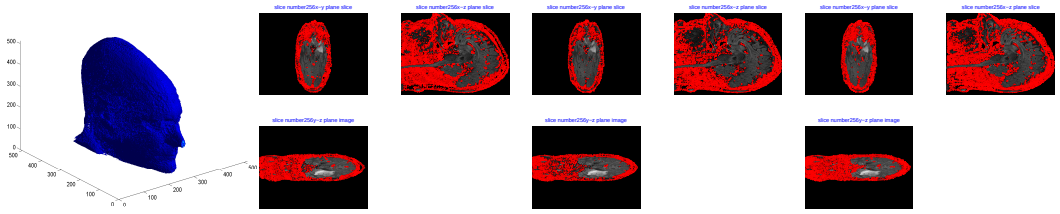


(b) Successfully reached relative residual equal to $5 * 10^{-1}$ after 2 iterations for the RHAMs and 1000 iterations for TM.



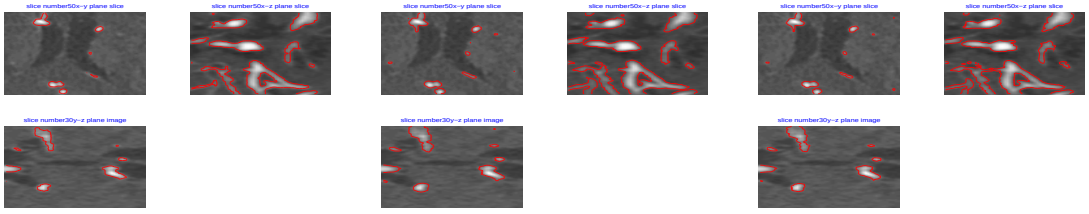
(c) Successfully reached relative residual equal to $5 * 10^{-3}$ after 4 iterations for the RHAMs and 28 iterations for TM.

Figure 4.15: Test Set-3 Example 2



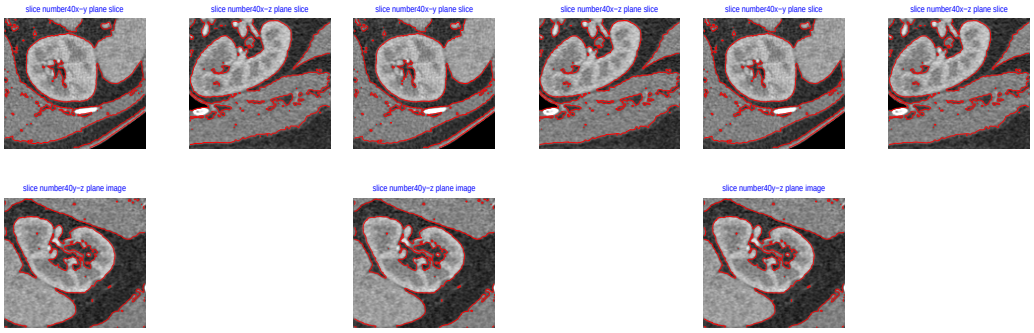
(a) Given image (b) Slices in x,y and z direction for TM (c) Slices in x,y and z direction for RHAM1 (d) Slices in x,y and z direction for RHAM2

Figure 4.16: Test Set-3 Example 3. Successfully reached relative residual equal to 10^{-1} after 2 iterations for the RHAMs and 1000 iterations for TM. Due to the size of the image TM CPU time was a few hours which is a great improvement with the RHAM which only required 5 minutes in this case.



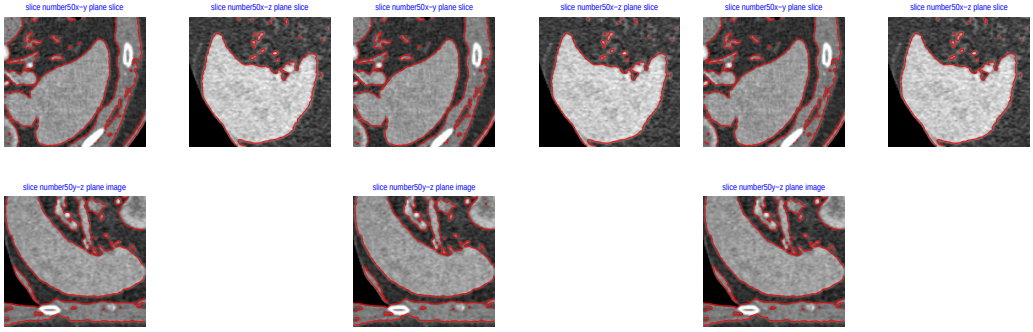
(a) Slices in x,y and z direction for TM (b) Slices in x,y and z direction for RHAM1 (c) Slices in x,y and z direction for RHAM2

Figure 4.17: Test Set-3 Example 4. Successfully reached relative residual equal to 10^{-1} by segmenting the vessels after 2 iterations for the RHAMs and for TM more than 100 iterations are needed.



(a) Slices in x,y and z direction for TM (b) Slices in x,y and z direction for RHAM1 (c) Slices in x,y and z direction for RHAM2

Figure 4.18: Test Set-3 Example 5. Successfully reached relative residual equal to 10^{-1} by segmenting a CT image near the kidney after 2 iterations for the RHAMs and for TM more than 100 iterations are needed.



(a) Slices in x,y and z direction for TM

(b) Slices in x,y and z direction for RHAM1

(c) Slices in x,y and z direction for RHAM2

Figure 4.19: Test Set-3 Example 6. Successfully reached relative residual equal to 10^{-1} by segmenting a CT image near the spleen after 3 iterations for the RHAMs and for TM more than 100 iterations are needed.

Table 4.1: Test Set-1: **Gaussian** noise applied to the image.
Required iterations for different methods (with $\Delta t = 0.01$) to achieve the PSNR values obtained from RHAM1 and RHAM2 by use of stopping criteria $PSNR(u_n) < PSNR(u_{n-1})$. The ✓ applies for cases when the TM method can have the same PSNR as RHAM, otherwise column four gives the maximum PSNR reached with the TM method.

Image	Noise	TM	TM PSNR	RHAM1	RHAM2	PSNR
	10	872	✓	48	50	**33.60**
Lena	15	1524	31.40	63	65	**31.47**
	20	1653	30.01	67	69	**30.13**
	10	1112	✓	52	53	**33.95**
Peppers	15	1581	31.61	65	67	**31.72**
	20	1721	30.19	68	71	**30.20**
	10	1415	✓	82	88	**41.46**
Objects	15	1941	✓	98	99	**38.85**
	20	1895	✓	85	87	**36.50**
	10	1308	✓	71	75	**40.74**
Rocket	15	1781	✓	83	85	**37.87**
	20	1729	✓	78	81	**35.49**

Table 4.2: Test Set-1: **Gaussian** noise applied to the image.
CPU time for different methods (with $\Delta t = 0.01$) to achieve the PSNR values obtained from RHAM1 and RHAM2 by use of stopping criteria $PSNR(u_n) < PSNR(u_{n-1})$. The ✓ applies for cases when the TM method can have the same PSNR as RHAM, otherwise column four gives the maximum PSNR reached with the TM method.

Image	Noise	TM	TM PSNR	RHAM1	RHAM2	PSNR
	10	80.42	✓	18.01	17.16	**33.6**
Lena	15	120.80	31.40	21.73	22.43	**31.47**
	20	151.04	30.01	23.46	23.68	**30.13**
	10	797	✓	19.68	18.42	**33.95**
Peppers	15	120.43	31.61	22.74	23.25	**31.72**
	20	150.82	30.19	20.84	24.32	**30.20**
	10	115.83	✓	28.16	30.15	**41.46**
Objects	15	152.49	✓	34.27	33.08	**38.85**
	20	154.65	✓	24.89	30.13	**36.50**
	10	111.79	✓	24.18	26.37	**40.74**
Rocket	15	130.68	✓	27.86	29.35	**37.87**
	20	114.59	✓	27.58	27.84	**35.49**

Table 4.3: Test Set-1: **Uniform** random distributed noise.
Required iterations for different methods (with $\Delta t = 0.01$) to achieve the PSNR values obtained from RHAM1 and RHAM2 by use of stopping criterion $PSNR(u_n) < PSNR(u_{n-1})$.

Image	Noise	TM	RHAM1	RHAM2	PSNR
	10	1000	41	44	**36.08**
Lena	30	717	719	43	**24.1585**
	50	2441	2450	86	**19.8548**
	10	127	127	19	**33.4051**
Peppers	30	784	778	45	**24.2216**
	50	2705	2713	93	**19.8986**
	10	216	217	26	**34.0754**
Objects	30	1016	1017	65	**24.5201**
	50	2881	2881	127	**20.142**
	10	207	207	25	**34.0573**
Rocket	30	950	951	59	**24.5216**
	50	2729	2730	116	**20.0786**

Table 4.4: Test Set-1 **Uniform** random distributed noise.
CPU time for different methods (with $\Delta t = 0.01$) for to achieve the PSNR values obtained from RHAM1 and RHAM2 by use of stopping criterion $PSNR(u_n) < PSNR(u_{n-1})$.

Image	Noise	TM	RHAM1	RHAM2	PSNR
	10	0.8268	0.9672	0.1716	**33.3785**
Lena	30	5.382	5.1948	0.4212	**24.1585**
	50	19.4065	20.1553	0.8892	**19.8548**
	10	0.9984	1.092	0.1716	**33.4051**
Peppers	30	5.850	5.616	0.4212	**24.2216**
	50	20.2489	20.8105	0.7332	**19.8986**
	10	1.6380	1.6380	0.234	**34.0754**
Objects	30	7.8156	8.4864	0.5460	**24.5201**
	50	21.7309	23.1817	0.9828	**20.142**
	10	1.2948	1.6692	0.2496	**34.0573**
Rocket	30	6.9888	7.8624	0.4836	**24.5216**
	50	20.8885	22.1677	0.8736	**20.0786**

Table 4.5: Test Set-1: AOS with **Gaussian** noise applied to the image to be compared with Tables 4.1-4.2.
Required iterations for the AOS method (with $\Delta t = 0.01$, 0.1 and 1) to achieve the PSNR values obtained from RHAM1 and RHAM2. The bold numbers in the PSNR columnsin table (4.5) shows that the PSNR of both AOS and the RHAMs have been the same, the rest of non bold PSNRs is the maximum that the AOS method could achieve.

		$\Delta t = .01$			$\Delta t = .1$			$\Delta t = 1$		
Image	Noise	CPU	No	PSNR	CPU	No	PSNR	CPU	No	PSNR
	10	86.04	867	33.44	9.18	91	33.36	1.32	13	33.26
Lena	15	130.26	1321	31.31	12.41	127	31.30	1.66	17	31.17
	20	147.48	1484	29.93	15.21	152	29.90	1.90	18	29.84
	10	77.47	782	**33.95**	9.06	93	**33.95**	1.54	15	33.83
Peppers	15	109.04	1108	**31.72**	12.32	125	**31.72**	1.87	18	31.72
	20	126.31	1275	**30.20**	13.11	133	**30.20**	1.81	16	30.20
	10	124.84	1264	**41.46**	15.24	153	**41.46**	41.04	109	41.04
Objects	15	205.85	2083	**38.85**	24.64	249	**38.85**	4.55	45	3824
	20	226.31	2286	36.23	19.68	199	36.22	2.9	128	35.80
	10	113.17	1141	**40.74**	12.52	126	**40.74**	4.45	44	40.59
Rocket	15	163.55	1646	**37.87**	17.05	172	**37.87**	3.1	31	37.68
	20	163.87	1644	**35.49**	16.16	164	**35.49**	2.60	25	35.42

Table 4.6: Test Set-2: Comparison of RHAM with TM.
Comparison for CPU time recorded and the maximum iterations needed to have a given residual as shown in the respective picture between Time Marching, RHAM1 and RHAM2 methods.

$\Delta t = 0.01$	CPU time			No. of iterations used		
	TM	RHAM1	RHAM2	TM	RHAM1	RHAM2
Fig.4.7	143.07	4.74	4.75	1287	29	32
Fig.4.8	22.45	4.24	4.82	205	25	33
Fig.4.9	20.03	2.19	2.21	186	9	9
Fig.4.10	97.83	6.70	7.50	929	46	53
Fig.4.11	49.48	7.22	6.45	467	49	47

Table 4.7: Test Set-2: Comparison of RHAM with AOS.
CPU time recorded for the iteration needed to have a given residual as shown in the respective picture.

CPU time	$\Delta t = 0.01$		$\Delta t = 0.1$		$\Delta t = 1$	
	AOS	RHAM1	AOS	RHAM1	AOS	RHAM1
Fig.4.12	593.73	22.67	47.53	10.17	13.87	2.98
Fig.4.13	175.00	12.05	53.03	3.03	6.32	2.50

Table 4.8: Test Set-2: Comparison of RHAM with AOS.
Maximum iterations needed to have a given residual as shown in the respective picture.

No. of iter. used	$\Delta t = 0.01$		$\Delta t = 0.1$		$\Delta t = 1$	
	AOS	RHAM1	AOS	RHAM1	AOS	RHAM1
Fig.4.12	1558	59	139	39	40	4
Fig.4.13	496	31	152	4	17	2

Table 4.9: Test Set-2: Comparison of RHAM with TM.
Comparison for CPU time recorded and the maximum iteration needed to have a given residual as shown in the respective pictures between the Time Marching, RHAM1 and RHAM2 methods.

$\Delta t = 0.01$	CPU time			No. of iterations used		
	TM	RHAM1	RHAM2	TM	RHAM1	RHAM2
Fig.4.14	3.4015e+04	86.19	84.25	1000	2	2
Fig.4.15(a)	2.6985e+04	85.31	82.43	800	2	2
Fig.4.15(b)	2.6985e+04	85.31	85.31	1000	2	2
Fig.4.15(c)	9525.99	527.19	549.92	28	3	3

Chapter 5

A New Variational Model with Dual Level Set Functions for 2-D Selective Segmentation

As a brief note, the work presented in this chapter has already been published [129] in the *Journal of Communications in Computational Physics* under the title *A new variational model with dual level set functions for selective segmentation* and it is coauthored together with my supervisor Prof. Ke Chen. The model presented here is a selective segmentation model [129] using a dual level set variational formulation. This model may be regarded as an improved version of the Gout-Guyader [66] and Badshah-Chen [12] models which were designed for selective segmentation under geometrical constraints using an active contour approach.

5.1 Introduction

In the previous chapters we mentioned different techniques (or even detailed some of them) developed for image segmentation such as histogram analysis and thresholding [107, 146, 167], region growing [3, 188], edge detection active contours [10, 28, 83, 92], region based active contours [115, 37, 39, 132, 166]. These image segmentation models are useful for various applications when all features/objects in the whole image are to be segmented. This kind of segmentation can not answer the simple question: "Can we segment only one desired object among the others?". This question is common in medical imaging of a particular organ or CCTV monitoring of a subject.

Assume that $u_0(x, y)$ is the given image defined on the rectangular 2-D domain Ω. In a selective segmentation, we hope to detect the features of image u_0 that are defined in a closed domain and are closest to the geometrical points in a set $\mathcal{A} = \{(x_i^*, y_i^*) \in \Omega, \quad 1 \leq i \leq n_1\} \subset \Omega$ consisting of n_1 distinct points near the object boundary to be detected [67, 69]. The aim of a selective segmentation is to find an optimal contour $\Gamma_L \subset \Omega$ that represents a closed object and best approaches the points from the set $\mathcal{A}$ in some sense of minimal geometric distance. Figure 5.1 illustrates a simple example with two objects. In this figure Γ_G represents the boundaries of all objects in the image while c_1 and c_2 similar to Chan-Vese [37] are the mean intensity of foreground (representing all the object) and background, respectively.

Recent work by Gout and Guyader [66] and Badshah-Chen [12] proposed two different variational models for selective segmentation. The Gout-Guyader model [69, 66] is

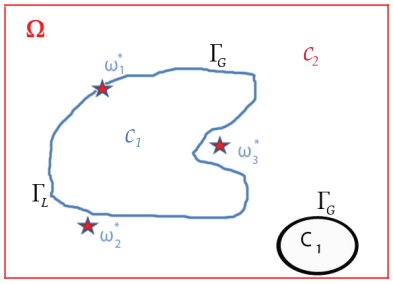
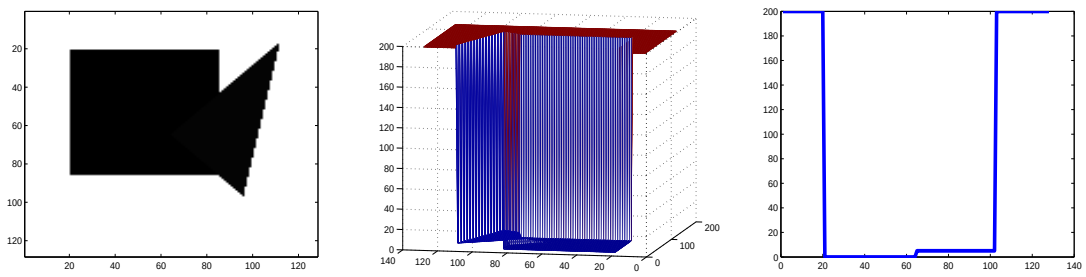


Figure 5.1: Illustration of notation of selective segmentation models with $n_1 = 3$ markers.

based on edge information of the object while the Badshah-Chen model [12] combines an edge based model with region based information. Both models are useful and can segment a range of images, but there are cases which appear too challenging for either. The main challenge for these models and in general for selective image segmentation problem is how to differentiate one feature from another, especially when two objects have similar or same intensities. Such an example is shown in Figure 5.2 where the intensities between the triangle and the rectangle in an artificial image have an extremely small difference and we might be interested in selecting only one of them. Another example is with medical images (CT and MRI) which often have less contrast in image intensities.



(a) (b) (c)

Figure 5.2: (a) An image of a 2-D triangle over a rectangle with small intensity difference; (b) Plot of intensities of the image; (c) Plane cut view through the vertical middle of the image.

To further improve on the Gout-Guyader [66] and Badshah-Chen [12] work a dual level set selective segmentation method [129] was introduced. The method aims to segment all objects with one level set function (global) and the selected object, which is the closest to the geometric constraints (markers), with another level set (local). The notation we will use is ϕ_G for the global function and ϕ_L for the local function. The model is a combination of edge detection, markers distance function and active contour without edges. Experimental results show that our model is more robust than previous work. We remark that this model may be further speeded up by using fast marching methods developed by Sethian [148], which were later extended in Farcadel-Guyader-Gout [57], through evolving boundary contours in a neighbourhood of the underlying zero level set curves. Main references for this chapter are [121, 92, 149, 38, 66, 12].

Before proceeding, we give an early clue for the comparison between Badshah-Chen

[12] and the new model [129] presented shortly. The bottom two plots of Figure 5.3 show correctly segmented results for dual level set selective segmentation, which solves the problem from Figure 5.2.

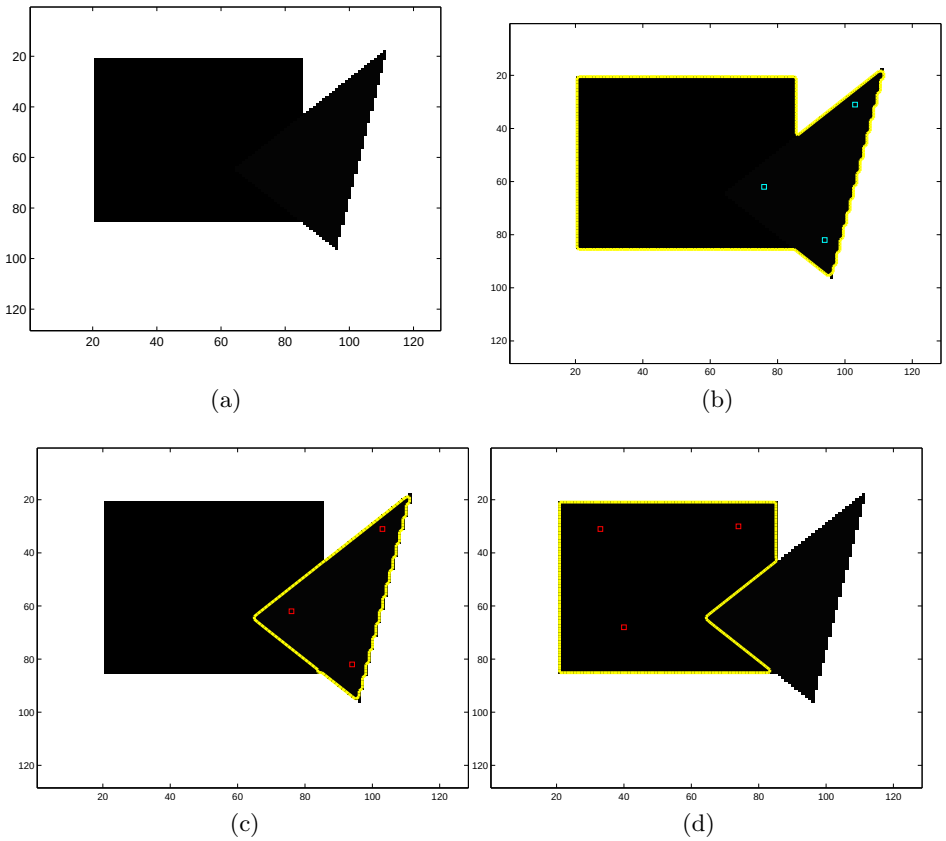


Figure 5.3: Comparison of (c) by the new dual level set selective segmentation model [129] and (b) by the Badshah-Chen model [12]. (a) given image; (b) incorrect result by the Badshah-Chen model; (c) correct selection of the triangle by the new dual level set model; (d) correct selection of the rectangle without triangle corner by the dual level set model.

The rest of the chapter is organized as follows. In Section §5.2 we give a review of the related model, Badshah-Chen [12], based on Gout-Guyader work [66]. Then in Section §5.3 we present the proposed dual level set selective segmentation model and derive the Euler-Lagrange equation. In Section §5.4 we describe an additive operator splitting method for solving the PDEs. In Section §5.5 we give some experimental results to further illustrate the improved robustness of our new model. Section §5.6 concludes the chapter.

5.2 Review of Existing Variational Selective Segmentation Models [12, 66]

Directly related to this work are the Chan-Vese [37] and Li-Xu-Gui-Fox [92] models that are already reviewed in Chapter 3. The above models are examples of global segmentation methods. To tackle the task of selective segmentation, these global models

are inadequate even if we start an initial contour from within an interested object. Below we review the selective model by Badshah-Chen [12] which is based on the Gout-Guyader model [66, 69].

For a given image $u_0(x, y)$ defined on the rectangular domain Ω, the aim of the model is to detect the features of image u_0 that are defined in a closed domain and be the closest to the geometrical points in a set $\mathcal{A}$ consisting of n_1 distinct points near the object boundary to be detected [67, 66, 69]. Let Γ denote the boundary that separates the two regions, Ω_1 the object of interest and $\Omega_2 = \Omega \backslash \overline{\Omega_1}$ the rest of the image.

The Gou-Guyader model [69] combines the geodesic active contour model [28] with the geometrical constraint of being close to $\mathcal{A}$. An edge detector function g is used to stop the evolving curve on edges of the objects in an image. The function g can be defined as

$$g(w) = \frac{1}{1 + w^2}, \tag{5.1}$$

or in other forms which can be found in [28, 38, 111]. Clearly $g(|\nabla u_0(x, y)|)$ is zero on edges in an image where $w = |\nabla u_0(x, y)|$ is large and one in flat regions where w is small. To help stop the evolving curve when approaching the points from set $\mathcal{A}$ a distance function d will be required. The distance function d is defined in the following way by Gout-Guyader [69]:

$$d(x, y) = \text{distance}((x, y), A) = \prod_{i=1}^{n_1} \left(1 - e^{-\frac{(x - x_i^*)^2}{2\tau^2}} e^{-\frac{(y - y_i^*)^2}{2\tau^2}}\right), \tag{5.2}$$

for all $(x, y) \in \Omega$. Other distance functions from Section §2.6.2 have been applied in different realizations, see [67, 179]. Clearly d acts locally and will be approximately 0 in the neighborhood of points of $\mathcal{A}$. Gout-Guyader model [66, 69] aims to find a contour Γ such that $d \simeq 0$ or $g \simeq 0$ along it. They proposed the following model for this purpose

$$\min_{\Gamma} F(\Gamma) = \int_{\Gamma} d(x, y) g(|\nabla u_0(x, y)|) ds. \tag{5.3}$$

The contour Γ will stop at local minima where $d \simeq 0$ (in the neighborhood of points for $\mathcal{A}$) or $g \simeq 0$ (near object boundaries).

On the other hand the Badshah-Chen [12] model improves on (5.3) by proposing the following model:

$$\min_{\phi(x,y), c_1, c_2} F(\phi(x, y), c_1, c_2),$$

where

$$\begin{aligned} F(\Gamma, c_1, c_2) = \; & \mu \int_{\Gamma} d(x, y) g(|\nabla u_0(x, y)|) ds + \\ & \lambda_1 \int_{\text{inside}(\Gamma)} |u_0(x, y) - c_1|^2 dx dy + \lambda_2 \int_{\text{outside}(\Gamma)} |u_0(x, y) - c_2|^2 dx dy, \end{aligned}$$

essentially adding two region-based terms weighted by λ_1 and λ_2 to equation (5.3). The

level set formulation of the functional (5.4) is:

$$\min_{\phi(x,y),c_1,c_2} F(\phi(x,y),c_1,c_2) = \mu \int_\Omega d(x,y) g(|\nabla u_0(x,y)|)|\nabla H(\phi(x,y))|dxdy + \lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H(\phi(x,y))dxdy + \lambda_2 \int_\Omega |u_0(x,y)-c_2|^2 (1-H(\phi(x,y)))dxdy, \quad (5.4)$$

where H is the Heaviside function.

Since the Heaviside function is not differentiable at the origin, we consider the regularized version of H denoted by H_ϵ and the corresponding δ by δ_ϵ as explained in Subsection §3.4.8. Then the minimization problem (5.4) becomes

$$\min_{\phi(x,y),c_1,c_2} F_\epsilon(\phi(x,y),c_1,c_2) = \mu \int_\Omega W\delta_\epsilon(\phi(x,y))|\nabla\phi(x,y)|dxdy + \lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H_\epsilon(\phi(x,y))dxdy + \lambda_2 \int_\Omega |u_0(x,y)-c_2|^2 (1-H_\epsilon(\phi(x,y)))dxdy, \quad (5.5)$$

where $W = d(x,y)g(|\nabla u_0(x,y)|)$. Keeping $\phi(x,y)$ fixed and minimizing with respect to c_1 and c_2, one gets the following equations for computing c_1 and c_2

$$c_1(\phi(x,y)) = \frac{\int_\Omega u_0(x,y)H_\epsilon(\phi(x,y))dxdy}{\int_\Omega H_\epsilon(\phi(x,y))dxdy} \quad (5.6)$$

if $\int_\Omega H_\epsilon(\phi(x,y))dxdy > 0$ (i.e if the curve has a nonempty interior in Ω), and

$$c_2(\phi(x,y)) = \frac{\int_\Omega u_0(x,y)(1-H_\epsilon(\phi(x,y)))dxdy}{\int_\Omega (1-H_\epsilon(\phi(x,y)))dxdy} \quad (5.7)$$

if $\int_\Omega (1-H_\epsilon(\phi(x,y)))dxdy > 0$ (i.e if the curve has a nonempty exterior in Ω). Finally keeping c_1 and c_2 fixed, one can minimize (5.5) with respect to $\phi(x,y)$ and derive the following Euler-Lagrange equation

$$\delta_\epsilon(\phi)\mu\nabla\cdot\left(W\frac{\nabla\phi}{|\nabla\phi|}\right) - \delta_\epsilon(\phi)(\lambda_1(u_0(x,y)-c_1)^2 - \lambda_2(u_0(x,y)-c_2)^2) = 0, \quad \text{in } \Omega \quad (5.8)$$

$$\text{with } W\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} = 0, \quad \text{on } \partial\Omega.$$

Even though this model improves the Gout-Guyader [69] model as shown in the Badshah-Chen [12] paper there are still cases when the models will fail and for this reason new models are needed.

5.3 Dual Level Set Selective Segmentation Variational Model

The dual level set selective segmentation model [129] is motivated by the fact that the region-based terms in Badshah-Chen model [12] essentially carry out a global segmen-

tation while the region-based terms are required for local segmentation, i.e. selection. A closer look at the generalized Badshah-Chen model (5.4) reveals that the model might tend to include inside the contour Γ unwanted objects which have similar intensity values c_1 to the selection object, and fail in this way. The other fitting value c_2 keeps changing as it is not well-defined during an evolving process. This suggests that the model might not have a unique solution. To this end, a dual level set approach is proposed [129], where ϕ_G and ϕ_L are respectively used to carry out global and local segmentation.

Given image $u_0(x,y)$, we denote $\Gamma_G = \partial\Omega_G$ in Ω the global evolving curve for locating all features Ω_G of image u_0 and the desired selective curve by $\Gamma_L = \partial\Omega_L$ in Ω, where we naturally assume Ω_L is contained in Ω_G. Thus we have $\text{inside}(\Gamma_L) = \Omega_L, \text{outside}(\Gamma_L) = \Omega\backslash\overline{\Omega_L}$, $\text{inside}(\Gamma_G) = \Omega_G$, $\text{outside}(\Gamma_G) = \Omega\backslash\overline{\Omega_G}$. The two zero level set functions $\phi_L(x,y)$ and $\phi_G(x,y)$ are defined such that

$$\begin{cases} \Gamma_L = \partial\Omega_L = \{(x,y) \in \Omega_L \;\Big|\; \phi_L(x,y) = 0\} \\ \text{inside}(\Gamma_L) = \Omega_L = \{(x,y) \in \Omega_L \;\Big|\; \phi_L(x,y) > 0\} \\ \text{outside}(\Gamma_L) = \Omega\backslash\overline{\Omega_L} = \{(x,y) \in \Omega_L \;\Big|\; \phi_L(x,y) < 0\}, \end{cases}$$

$$\begin{cases} \Gamma_G = \partial\Omega_G = \{(x,y) \in \Omega \;\Big|\; \phi_G(x,y) = 0\} \\ \text{inside}(\Gamma_G) = \Omega_G = \{(x,y) \in \Omega \;\Big|\; \phi_G(x,y) > 0\} \\ \text{outside}(\Gamma_G) = \Omega\backslash\overline{\Omega_G} = \{(x,y) \in \Omega \;\Big|\; \phi_G(x,y) < 0\}. \end{cases}$$

In this way we replace the unknown quantities Γ_L by ϕ_L and Γ_G by ϕ_G shortly. To explore possible advantages of having an enlarged domain of Ω_G within a distance of γ away, we define

$$\Omega_{G,\gamma} = \{(x,y) \in \Omega \;\big|\; \phi_G(x,y) > -\gamma\},$$

where the parameter $\gamma \geq 0$ will be taken as 0 or 3. Note $\Omega_L \subset \Omega_G \subseteq \Omega_{G,\gamma} \subset \Omega$.

Then realizing the idea of looking at all features of Ω_G in the whole image domain Ω and the selective features Ω_L in the local domain Ω_G, our new variational model is the following

$$\begin{aligned} \min_{\Gamma_L,\Gamma_G,c_1,c_2} \; & F(\Gamma_L,\Gamma_G,c_1,c_2) = \\ & \mu_1 \int_{\Gamma_L} d(x,y) g(|\nabla u_0(x,y)|) ds \; + \; \mu_2 \int_{\Gamma_G} g(|\nabla u_0(x,y)|) ds \; + \\ & \lambda_{1\text{G}} \int_{\text{inside}(\Gamma_G)} |u_0(x,y) - c_1|^2 dxdy \; + \\ & \lambda_{2\text{G}} \int_{\text{outside}(\Gamma_G)} |u_0(x,y) - c_2|^2 dxdy \; + \\ & \lambda_1 \int_{\text{inside}(\Gamma_L)} |u_0(x,y) - c_1|^2 dxdy \; + \\ & \lambda_2 \int_{\text{outside}(\Gamma_L)\cap\text{inside}(\Gamma_G)} |u_0(x,y) - c_1|^2 dxdy \; + \\ & \lambda_3 \int_{\text{outside}(\Gamma_L)\cap\text{outside}(\Gamma_G)} |u_0(x,y) - c_2|^2 dxdy, \end{aligned} \tag{5.9}$$

where

$$g(|\nabla u_0(x,y)|) = \frac{1}{1+|\nabla G_\sigma(x,y) * u_0(x,y)|^2}, \tag{5.10}$$

μ_1 , μ_2 , $\lambda_{1\mathrm{G}}$, $\lambda_{2\mathrm{G}}$, λ_1, λ_2, λ_3 are all positive parameters, $d(x,y)$ is a distance function from the given geometric markers in set $\mathcal{A}$ as defined in [12]. Here $G_\sigma(x,y) * u_0(x,y)$ is a smooth version of $u_0(x,y)$, with Gaussian $G_\sigma(x,y) = \sigma^{-1/2} e^{-|x^2+y^2|/4\sigma}$, to deal with possible noise (in our experiments for the image with strong noise, σ is taken $1/2$).

To avoid re-initialization new terms similar to $P(\phi)$ in Li-Xu-Gui-Fox [92] model, which is already reviewed in Chapter 3, will automatically scale both level set functions $\phi_L(x,y)$ and $\phi_G(x,y)$. We emphasize that the first term in the introduced minimization functional constrains the search domain for computing the weighted length of Γ_L to $\Omega_{G,\gamma}$ instead of Ω and we have

$$\begin{aligned}
\int_{\Gamma_L} d(x,y)g(|\nabla u_0(x,y)|)ds &= \int_{\Omega} d(x,y)g(|\nabla u_0(x,y)|)|\nabla H(\phi_L(x,y))dxdy \\
&= \int_{\Omega_{G,\gamma}} d(x,y)g(|\nabla u_0(x,y)|)|\nabla H(\phi_L(x,y))dxdy \\
&= \int_{\Omega} d(x,y)g(|\nabla u_0(x,y)|)|\nabla H(\phi_L(x,y))|H(\phi_G(x,y)+\gamma)dxdy.
\end{aligned}$$

With a regularized Heaviside, equation (5.9) can be rewritten as

$$\begin{aligned}
&\min_{\phi_L(x,y),\phi_G(x,y),c_1,c_2} F_\epsilon(\phi_L(x,y),\phi_G(x,y)c_1,c_2) \; = \\
&\quad \mu_1 \int_\Omega d(x,y)g(|\nabla u_0(x,y)|)\delta_\epsilon(\phi_L(x,y))|\nabla\phi_L(x,y)|H_\epsilon(\phi_G(x,y)+\gamma)dxdy \; + \\
&\quad \frac{\mu_{\mathrm{L}}}{2}\int_\Omega (|\nabla\phi_L(x,y)|-1)^2 dxdy \; + \\
&\quad \mu_2 \int_\Omega g(|\nabla u_0(x,y)|)\delta_\epsilon(\phi_G(x,y))|\nabla\phi_G(x,y)|dxdy \; + \\
&\quad \frac{\mu_{\mathrm{G}}}{2}\int_\Omega (|\nabla\phi_G(x,y)|-1)^2 dxdy \; + \\
&\quad \lambda_{1\mathrm{G}}\int_\Omega |u_0(x,y)-c_1|^2 H_\epsilon(\phi_G(x,y)dxdy \; + \\
&\quad \lambda_{2\mathrm{G}}\int_\Omega |u_0(x,y)-c_2|^2 (1-H_\epsilon(\phi_G(x,y))dxdy \; + \\
&\quad \lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H_\epsilon(\phi_L(x,y)dxdy \; + \\
&\quad \lambda_2 \int_\Omega |u_0(x,y)-c_1|^2 (1-H_\epsilon(\phi_L(x,y))H(\phi_G(x,y)dxdy \; + \\
&\quad \lambda_3 \int_\Omega |u_0(x,y)-c_2|^2 (1-H_\epsilon(\phi_L(x,y))(1-H_\epsilon(\phi_G(x,y))dxdy.
\end{aligned} \tag{5.11}$$

Here μ_L and μ_G are positive. For brevity, we use d, u_0, ϕ_L, ϕ_G to denote $d(x,y)$, $u_0(x,y)$, $\phi_L(x,y)$ and $\phi_G(x,y)$, respectively.

Keeping ϕ fixed and minimizing with respect to c_1 and c_2, we have the following equations for computing c_1 and c_2:

$$c_1 = \frac{\lambda_{1\mathrm{G}}\int_\Omega u_0 H_\epsilon(\phi_G)dxdy + \lambda_1\int_\Omega u_0 H_\epsilon(\phi_L)dxdy + \lambda_2\int_\Omega u_0(1-H_\epsilon(\phi_L))H_\epsilon(\phi_G)dxdy}{\lambda_{1\mathrm{G}}\int_\Omega H_\epsilon(\phi_G)dxdy + \lambda_1\int_\Omega H_\epsilon(\phi_L)dxdy + \lambda_2\int_\Omega (1-H_\epsilon(\phi_L))H_\epsilon(\phi_G)dxdy}, \tag{5.12}$$

$$c_2 = \frac{\lambda_{2G}\int_\Omega u_0(1-H_\epsilon(\phi_G))dxdy + \lambda_3\int_\Omega u_0(1-H_\epsilon(\phi_L))(1-H_\epsilon(\phi_G))dxdy}{\lambda_{2G}\int_\Omega(1-H_\epsilon(\phi_G))dxdy + \lambda_3\int_\Omega(1-H_\epsilon(\phi_L))(1-H_\epsilon(\phi_G))dxdy}, \quad (5.13)$$

if we assume that $\phi_G(x,y)$ has neither empty interior nor empty exterior.

Keeping c_1 and c_2 fixed, we minimize (5.11) with respect to $\phi_L(x,y)$ and $\phi_G(x,y)$. We first minimize F_ϵ with respect to ϕ_L by using the Gâteaux derivatives to find the first variation of the functional F_ϵ with respect to ϕ_L

$$\lim_{h\to 0}\frac{d}{dh}\Big(F_\epsilon(\phi_L+h\psi,c_1,c_2)\Big)=0$$

$$\begin{aligned}
\Rightarrow \quad & \mu_1\frac{d}{dh}\int_\Omega dg(|\nabla u_0|)\delta_\epsilon(\phi_L+h\psi)|\nabla(\phi_L+h\psi)|H_\epsilon(\phi_G+\gamma)dxdy\Big|_{h=0} + \\
& \frac{\mu_L}{2}\frac{d}{dh}\int_\Omega(|\nabla\phi_L+h\psi|-1)^2dxdy\Big|_{h=0} + \\
& \frac{d}{dh}\int_\Omega\Big(\lambda_1|u_0-c_1|^2H_\varepsilon(\phi_L+h\psi)+\lambda_2|u_0-c_1|^2(1-H_\varepsilon(\phi_L+h\psi))H_\varepsilon(\phi_G)+ \\
& \lambda_3|u_0-c_2|^2(1-H_\varepsilon(\phi_G))(1-H_\varepsilon(\phi_L+h\psi))\Big)dxdy\Big|_{h=0}=0 \\
\Rightarrow \quad & \mu_1\int_\Omega dg(|\nabla u_0|)H_\epsilon(\phi_G+\gamma)\Big(|\nabla(\phi_L+h\psi)|\frac{d}{dh}\delta_\epsilon(\phi_L+h\psi)+ \\
& \delta_\epsilon(\phi_L+h\psi)\frac{d}{dh}|\nabla(\phi_L+h\psi)|\Big)dxdy\Big|_{h=0} + \\
& \frac{\mu_L}{2}\int_\Omega 2(|\nabla\phi_L+h\psi|-1)\frac{d}{dh}(|\nabla\phi_L+h\psi|)dxdy\Big|_{h=0} + \\
& +\int_\Omega\Big(\lambda_1(u_0(x,y)-c_1)^2\delta_\varepsilon(\phi_L+h\psi)\psi- \\
& \lambda_2(u_0(x,y)-c_1)^2H_\varepsilon(\phi_G)\delta_\varepsilon(\phi_L+h\psi)\psi- \\
& \lambda_3(u_0(x,y)-c_2)^2(1-H_\varepsilon(\phi_G))\delta_\varepsilon(\phi_L+h\psi)\psi\Big)dxdy\Big|_{h=0}=0
\end{aligned}$$

i.e. with $W = dg(|\nabla u_0|)$

$$\begin{aligned}
& \mu_1\textstyle\int_\Omega W H_\epsilon(\phi_G+\gamma)\Big(\delta'_\epsilon(\phi_L)|\nabla\phi_L|\psi+\delta_\epsilon(\phi_L)\frac{\nabla\phi_L}{|\nabla\phi_L|}\cdot\nabla\psi\Big)dxdy+ \\
& \mu_L\textstyle\int_\Omega(|\nabla\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}\cdot\nabla\psi dxdy+ \\
& \textstyle\int_\Omega\delta_\epsilon(\phi_L)\Big(\lambda_1(u_0-c_1)^2-\lambda_2(u_0-c_1)^2H_\varepsilon(\phi_G)-\lambda_3(u_0-c_2)^2(1-H_\varepsilon(\phi_G))\Big)\psi dxdy=0
\end{aligned}$$

or

$$\begin{aligned}
& \mu_1\textstyle\int_\Omega W H_\epsilon(\phi_G+\gamma)\delta'_\epsilon(\phi_L)|\nabla\phi_L|\psi dxdy+\ \mu_1\int_\Omega W H_\epsilon(\phi_G+\gamma)\delta_\epsilon(\phi_L)\frac{\nabla\phi_L}{|\nabla\phi_L|}\cdot\nabla\psi dxdy+ \\
& \mu_L\textstyle\int_\Omega(|\nabla\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}\cdot\nabla\psi dxdy+ \\
& \textstyle\int_\Omega\delta_\epsilon(\phi_L)\Big(\lambda_1(u_0-c_1)^2-\lambda_2(u_0-c_1)^2H_\varepsilon(\phi_G)-\lambda_3(u_0-c_2)^2(1-H_\varepsilon(\phi_G))\Big)\psi dxdy=0
\end{aligned}$$

where ψ is a test function of the same type as ϕ_L. Applying Green's identity

$$\int_\Omega v\nabla\cdot\vec{w}dx = -\int_\Omega\nabla v\cdot\vec{w}dx+\int_{\partial\Omega}v\vec{w}\cdot\vec{n}ds$$

to the middle two integrals of (5.3) with $\psi = v$, by taking $\vec{w} = WH_\epsilon(\phi_G)+\gamma)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\nabla\phi_L$ for the second integral and $\vec{w} = (|\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}$ for the third integral, we rewrite them respectively as

$$\begin{aligned}\int_\Omega WH_\epsilon(\phi_G+\gamma)\delta_\epsilon(\phi_L)\frac{\nabla\phi_L}{|\nabla\phi_L|}\cdot\nabla\psi dx = & -\int_\Omega \nabla\cdot\left(WH_\epsilon(\phi_G+\gamma)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\nabla\phi\right)\psi dxdy+\\ & \int_{\partial\Omega} WH_\epsilon(\phi_G+\gamma)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}}\psi ds,\end{aligned}$$

$$\begin{aligned}\int_\Omega (|\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}\cdot\nabla\psi dx &= -\int_\Omega \nabla\cdot\left((|\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy+\\ & \int_{\partial\Omega}(|\phi_L|-1)\frac{1}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}}\psi ds,\end{aligned}$$

where $\nabla\phi_L\cdot\vec{n} = \frac{\partial\phi_L}{\partial\vec{n}}$. Thus equation (5.3) becomes

$$\begin{cases}\mu_1\int_\Omega WH_\epsilon(\phi_G+\gamma)\delta'_\epsilon(\phi_L)|\nabla\phi_L|\psi dxdy \ +\mu_1\int_{\partial\Omega} WH_\epsilon(\phi_G+\gamma)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}}\psi ds\\ -\mu_1\int_\Omega\nabla\cdot\left(\delta_\epsilon(\phi_L)WH_\epsilon(\phi_G+\gamma)\frac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy \ +\\ \mu_{\mathrm{L}}\int_{\partial\Omega}(|\phi_L|-1)\frac{1}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}}\psi ds-\mu_{\mathrm{L}}\int_\Omega\nabla\cdot\left((|\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy \ +\\ \int_\Omega\delta_\epsilon(\phi_L)\Big(\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_1)^2H_\varepsilon(\phi_G)\ -\\ \qquad\lambda_3(u_0(x,y)-c_2)^2(1-H_\varepsilon(\phi_G))\Big)\psi dxdy=0,\end{cases}$$

$$\Rightarrow\quad\begin{cases}\mu_1\int_\Omega WH_\epsilon(\phi_G+\gamma)\delta'_\epsilon(\phi_L)|\nabla\phi|\psi dxdy \ +\\ \mu_1\int_{\partial\Omega} WH_\epsilon(\phi_G+\gamma)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}}\psi ds \ -\\ \mu_1\int_\Omega\delta_\epsilon(\phi_L)\nabla\cdot\left(WH_\epsilon(\phi_G+\gamma)\frac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy \ -\\ \mu_1\int_\Omega\delta'_\epsilon(\phi_L)WH_\epsilon(\phi_G+\gamma)\nabla\phi_L\cdot\frac{\nabla\phi_L}{|\nabla\phi_L|}\psi dxdy+\\ \mu_{\mathrm{L}}\int_{\partial\Omega}(|\phi_L|-1)\frac{1}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}}\psi ds-\mu_{\mathrm{L}}\int_\Omega\nabla\cdot\left((|\phi_L|-1)\frac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy \ +\\ \int_\Omega\delta_\epsilon(\phi_L)\Big(\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_1)^2H_\varepsilon(\phi_G)\ -\\ \qquad\lambda_3(u_0(x,y)-c_2)^2(1-H_\varepsilon(\phi_G))\Big)\psi dxdy \ =\ 0.\end{cases}$$

This gives

$$\begin{cases} -\mu_1 \int_\Omega \delta_\epsilon(\phi_L)\nabla\cdot\left(WH_\epsilon(\phi_G+\gamma)\dfrac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy - \\ \mu_{\mathrm{L}} \int_\Omega \nabla\cdot\left((|\phi_L|-1)\dfrac{\nabla\phi_L}{|\nabla\phi_L|}\right)\psi dxdy + \\ \mu_1 \int_{\partial\Omega} WH_\epsilon(\phi_G+\gamma)\dfrac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\dfrac{\partial\phi_L}{\partial\vec{n}}\psi ds + \mu_{\mathrm{L}}\int_{\partial\Omega}\psi(|\phi_L|-1)\dfrac{1}{|\nabla\phi_L|}\dfrac{\partial\phi_L}{\partial\vec{n}}\psi ds + \\ \int_\Omega \delta_\epsilon(\phi_L)\Big(\lambda_1(u_0(x,y)-c_1)^2 - \lambda_2(u_0(x,y)-c_1)^2 H_\varepsilon(\phi_G) - \\ \qquad \lambda_3(u_0(x,y)-c_2)^2(1-H_\varepsilon(\phi_G))\Big)\psi dxdy \;=\; 0. \end{cases}$$

As this holds for all test functions ψ, we have the following Euler-Lagrange equation for ϕ_L:

$$\begin{cases} \mu_1\delta_\epsilon(\phi_L)\nabla\cdot\left(WH_\epsilon(\phi_G+\gamma)\dfrac{\nabla\phi_L}{|\nabla\phi_L|}\right) + \mu_L\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_L|})\nabla\phi_L\right) + \\ \delta_\epsilon(\phi_L)\Big(-\lambda_1(u_0(x,y)-c_1)^2 + \lambda_2(u_0(x,y)-c_1)^2 H_\varepsilon(\phi_G) + \\ \quad \lambda_3(u_0(x,y)-c_2)^2(1-H_\varepsilon(\phi_G))\Big) = 0, \quad \text{in } \Omega \\ \frac{\partial\phi_L}{\partial\vec{n}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{5.14}$$

where boundary conditions $\mu_1 WH_\epsilon(\phi_G+\gamma)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}} = 0$ and $\mu_{\mathrm{L}}(|\phi_L|-1)\frac{1}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}} = 0$ reduce to Neumann boundary condition. In the same way we may derive the Euler-Lagrange equation for ϕ_G.

In equations for ϕ_G and ϕ_L, balloon terms such as $\alpha W|\nabla\phi_L|$ and $\alpha g(x,y)|\nabla\phi_G|$ respectively, can be added to speed up convergence. The final equations of ϕ_G and ϕ_L can be written in the form

$$\begin{cases} \mu_1\delta_\epsilon(\phi_L)\nabla\cdot\left(WH_\epsilon(\phi_G+\gamma)\dfrac{\nabla\phi_L}{|\nabla\phi_L|}\right) + \mu_L\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_L|})\nabla\phi_L\right) + \\ \delta_\epsilon(\phi_L)\Big(-\lambda_1(u_0(x,y)-c_1)^2 + \lambda_2(u_0(x,y)-c_1)^2 H_\varepsilon(\phi_G) + \\ \qquad \lambda_3(u_0(x,y)-c_2)^2(1-H_\varepsilon(\phi_G))\Big) \;+\; \alpha W(x,y)|\nabla\phi_L| = 0, \quad \text{in } \Omega \\ \frac{\partial\phi_L}{\partial\vec{n}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{5.15}$$

and

$$\begin{cases} \mu_2\delta_\epsilon(\phi_G)\nabla\cdot\left(g(x,y)\dfrac{\nabla\phi_G}{|\nabla\phi_G|}\right) + \mu_G\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_G|})\nabla\phi_G\right) + \\ \delta_\epsilon(\phi_G+\gamma)\Big(-\mu_1 W(x,y)\,|\nabla H_\varepsilon(\phi_L)|\Big) + \\ \delta_\epsilon(\phi_G)\Big(-\lambda_{1\mathrm{G}}(u_0(x,y)-c_1)^2 + \lambda_{2\mathrm{G}}(u_0(x,y)-c_2)^2 - \lambda_2(u_0(x,y)-c_1)^2(1-H(\phi_L)) \\ \qquad + \lambda_3(u_0(x,y)-c_2)^2(1-H(\phi_L))\Big) \;+\; \alpha g(x,y)|\nabla\phi_G| = 0, \quad \text{in } \Omega \\ \frac{\partial\phi_G}{\partial\vec{n}} = 0 \;\text{ on } \partial\Omega. \end{cases} \tag{5.16}$$

By freezing the nonlinear coefficients in equations (5.15) and (5.16) we get linearized

systems of equations which can be solved by a fixed point method. Since the drawback of this method is the computational cost of the associated linear system for large images, we develop a fast method similar to [171, 12, 69, 174].

5.4 An Additive Operator Splitting Algorithm

In the following we develop an additive operator splitting (AOS) method [104, 174] for (5.15) and (5.16), by considering first the following related parabolic equations:

$$\begin{cases} \phi_L(x,y,0) = \phi_L^0(x,y) \\ \dfrac{\partial \phi_L}{\partial t} = \mu_1 \delta_\epsilon(\phi_L) \nabla \cdot \left(W H_\epsilon(\phi_G + \gamma) \dfrac{\nabla \phi_L}{|\nabla \phi_L|} \right) + \mu_L \nabla \cdot \left((1 - \dfrac{1}{|\nabla \phi_L|}) \nabla \phi_L \right) + \\ \qquad \delta_\epsilon(\phi_L) \Big(- \lambda_1 (u_0(x,y) - c_1)^2 + \lambda_2 (u_0(x,y) - c_1)^2 H_\varepsilon(\phi_G) + \\ \qquad \lambda_3 (u_0(x,y) - c_2)^2 (1 - H_\varepsilon(\phi_G)) \Big) \; + \; \alpha W(x,y) |\nabla \phi_L|, \\ \dfrac{\partial \phi_L}{\partial \vec{n}} \Big|_{\partial \Omega} = 0, \end{cases} \tag{5.17}$$

$$\begin{cases} \phi_G(x,y,0) = \phi_G^0(x,y) \\ \dfrac{\partial \phi_G}{\partial t} = \mu_2 \delta_\epsilon(\phi_G) \nabla \cdot \left(g(x,y) \dfrac{\nabla \phi_G}{|\nabla \phi_G|} \right) + \mu_G \nabla \cdot \left((1 - \dfrac{1}{|\nabla \phi_G|}) \nabla \phi_G \right) + \\ \qquad \delta_\epsilon(\phi_G + \gamma) \Big(- \mu_1 W(x,y) \, |\nabla H_\varepsilon(\phi_L)| \Big) \; + \; \alpha g(x,y) |\nabla \phi_G| \; + \\ \qquad \delta_\epsilon(\phi_G) \Big(- \lambda_{1\mathrm{G}} (u_0(x,y) - c_1)^2 + \lambda_{2\mathrm{G}} (u_0(x,y) - c_2)^2 - \\ \qquad \lambda_2 (u_0(x,y) - c_1)^2 (1 - H(\phi_L)) + \lambda_3 (u_0(x,y) - c_2)^2 (1 - H(\phi_L)) \Big), \\ \dfrac{\partial \phi_G}{\partial \vec{n}} \Big|_{\partial \Omega} = 0. \end{cases} \tag{5.18}$$

Denoting

$$f_L = \delta_\epsilon(\phi_L) \Big(- \lambda_1 (u_0(x,y) - c_1)^2 + \lambda_2 (u_0(x,y) - c_1)^2 H_\varepsilon(\phi_G) + \lambda_3 (u_0(x,y) - c_2)^2 (1 - H_\varepsilon(\phi_G)) \Big) + \alpha W(x,y) |\nabla \phi_L|,$$

$$f_G = \delta_\epsilon(\phi_G + \gamma) \Big(- \mu_1 W(x,y) H_\varepsilon(\phi_L) \Big) + delta_\epsilon(\phi_G) \Big(- \lambda_{1\mathrm{G}} (u_0(x,y) - c_1)^2 + \lambda_{2\mathrm{G}} (u_0(x,y) - c_2)^2 - \lambda_2 (u_0(x,y) - c_1)^2 (1 - H(\phi_L)) + \lambda_3 (u_0(x,y) - c_2)^2 (1 - H(\phi_L)) \Big) + \alpha g(x,y) |\nabla \phi_G|$$

$$F_L = \frac{W H_\epsilon(\phi_G + \gamma)}{|\nabla \phi_L|}, \; F_G = \frac{g}{|\nabla \phi_G|}, \; E_L = 1 - \frac{1}{|\nabla \phi_L|}, \; E_G = 1 - \frac{1}{|\nabla \phi_G|},$$

the equations (5.18) and (5.17) can be written in the compact form:

$$\begin{cases} \dfrac{\partial \phi_L}{\partial t} = \mu_1 \delta_\epsilon(\phi_L)\nabla\cdot(F_L\nabla\phi_L) + \mu_{\rm L}\nabla\cdot(E_L\nabla\phi_L) + f_L \ = \\ \qquad \mu_1\delta_\epsilon(\phi_L)(\partial_x(F_L\partial_x\phi_L) + \partial_y(F_L\partial_y\phi_L)) + \mu_{\rm L}(\partial_x(E_L\partial_x\phi_L) + \partial_y(E_L\partial_y\phi_L)) + f_L, \\ \dfrac{\partial \phi_G}{\partial t} = \mu_2 \delta_\epsilon(\phi_G)\nabla\cdot(F_G\nabla\phi_G) + \mu_{\rm G}\nabla\cdot(E_G\nabla\phi_G) + f_G \ = \\ \qquad \mu_2\delta_\epsilon(\phi_G)(\partial_x(F_G\partial_x\phi_G) + \partial_y(F_G\partial_y\phi_G)) + \mu_{\rm G}(\partial_x(E_G\partial_x\phi_G) + \partial_y(E_G\partial_y\phi_G)) + f_G. \end{cases} \tag{5.19}$$

Since ϕ_G and ϕ_L depend on each other and the coefficients contain the nonlinearities, we have to iterate the above equations. Note that both equations in (5.19) are of similar self-adjoint forms. In the following it is sufficient to consider how to solve the second equation:

$$\begin{aligned} \frac{\partial \phi_G}{\partial t} = \mu_2\delta_\epsilon(\phi_G)(\partial_x(F_G\partial_x\phi_G) + \partial_y(F_G\partial_y\phi_G)) + \\ \mu_G(\partial_x(E_G\partial_x\phi_G) + \partial_y(E_G\partial_y\phi_G)) + f_G. \end{aligned} \tag{5.20}$$

Discretizing with spatial step size $h_1 = h_2 = h = 1$ and writing ϕ, F, E, after dropping the subscripts in ϕ_G, F_G, E_G, equation (5.20) leads to the semi-implicit equation in the x-coordinate direction

$$\begin{aligned} \frac{\phi_{i,j}^{k+1} - \phi_{i,j}^k}{2\triangle t} = \mu_2\delta_\epsilon(\phi_i)\bigg((\frac{F_{i,j}^k + F_{i+1,j}^k}{2h^2})(\phi_{i+1,j}^{k+1} - \phi_{i,j}^{k+1}) - (\frac{F_{i,j}^k + F_{i-1,j}^k}{2h^2})(\phi_{i,j}^{k+1} - \phi_{i-1,j}^{k+1})\bigg) + \\ \mu_G\bigg((\frac{E_{i,j}^k + E_{i+1,j}^k}{2h^2})(\phi_{i+1,j}^{k+1} - \phi_{i,j}^{k+1}) - (\frac{E_{i,j}^k + E_{i-1,j}^k}{2h^2})(\phi_{i,j}^{k+1} - \phi_{i-1,j}^{k+1})\bigg) + \frac{1}{2}f_{i,j}, \\ \Rightarrow \phi_{i,j}^{k+1} = \phi_{i,j}^k + 2\triangle t(w_1\phi_{i+1,j}^{k+1} - w_2\phi_{i,j}^{k+1} + w_3\phi_{i-1,j}^{k+1}) + \triangle t f_{i,j}, \end{aligned} \tag{5.21}$$

with

$$\begin{aligned} w_1 &= \mu_2\delta_\epsilon(\phi_{i,j})\frac{F_{i,j}^k + F_{i+1,j}^k}{2h^2} + \mu_G\frac{E_{i,j}^k + E_{i+1,j}^k}{2h^2}, \\ w_2 &= \mu_2\delta_\epsilon(\phi_{i,j})\frac{F_{i-1,j}^k + 2F_{i,j}^k + F_{i+1,j}^k}{2h^2} + \mu_G\frac{E_{i-1,j}^k + 2E_{i,j}^k + E_{i+1,j}^k}{2h^2}, \\ w_3 &= \mu_2\delta_\epsilon(\phi_{i,j})\frac{F_{i,j}^k + F_{i-1,j}^k}{2h^2} + \mu_G\frac{E_{i,j}^k + E_{i-1,j}^k}{2h^2}. \end{aligned} \tag{5.22}$$

Similarly, in the y-coordinate direction, one gets

$$\Rightarrow \phi_{i,j}^{k+1} = \phi_{i,j}^k + 2\triangle t(\bar{w}_1\phi_{i,j+1}^{k+1} - \bar{w}_2\phi_{i,j}^{k+1} + \bar{w}_3\phi_{i,j-1}^{k+1}) + \triangle t f_{i,j}. \tag{5.23}$$

In the spirit of AOS, we solve the decoupled system of equations (5.21) and (5.23) with time step $2\triangle t$ respectively in the x, y-direction and then average the two solutions with the result equivalent to solving a coupled semi-implicit system with time step $\triangle t$. In matrix notation, equations (5.21) and (5.23) can be written as:

$$(I - 2\triangle t A_l(\phi^k))\phi_l^{k+1} = \hat{f}^k, \text{ for } l = 1, 2: \quad \phi^{k+1} = \frac{1}{2}\sum_{l=1}^{2}\phi_l^{k+1}, \quad k = 0, 1, \ldots,$$

where $\hat{f}^k = \phi^k + \triangle t f^k$, I is the identity matrix and A_l is a tridiagonal matrix respectively for $l = 1, 2$ consisting of $\{w_1, -w_2, w_3\}$ and $\{\bar{w}_1, -\bar{w}_2, \bar{w}_3\}$ (adjusted at boundary

nodes).

5.5 Experimental Results

In order to illustrate the robustness of our new proposed model further numerical results will be applied in a range of artificial, synthetic and real images, with different types of contours and shapes. Our work will also be compared with Badshah-Chen [12] model. We will see that for problems which can be solved by both models, our method is less dependent on the choice of regularized Heaviside functions while there exist some cases where the latter method does not work.

In our numerical experiments, we generally choose two image sizes $n = 128, 256$ and the parameters as follows:

$\lambda_{1G} = 1$, $\lambda_{2G} = 1$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 1$, $\tau = 4$, $h = 1$ (the step space),
$\triangle t = 0.1$ or 0.01 (the time step), $\alpha = 0.001$,
$\mu_{\mathrm{L}} = 0.4$, $\mu_{\mathrm{G}} = 0.4$, $\mu_1 = n^2/10$, $\mu_2 = n^2/10$

(if a given image has no noise, then all μ parameters can be chosen smaller). The initial global level set, placed as a circle, has the form

$$\phi_G^0 = \sqrt{(x - x_G^0)^2 + (y - y_G^0)^2} - r_G^0,$$

where $(x_G^0,\ y_G^0)$ is the centre of the circle, usually at the center of Ω and $r_G^0 = n/5$ the radius, and the initial local level set is placed similarly as

$$\phi_L^0 = \sqrt{(x - x_L^0)^2 + (y - y_L^0)^2} - r_L^0,$$

where $(x_L^0,\ y_L^0)$ is the centre of the markers in set A and the radius r_L^0 is the minimum distance of the markers $r_L^0 = \min_{a \neq b} ||p_a - p_b||$, where $p_a, p_b \in A$; here $x_L^0 = \frac{\sum x\text{-comp of markers}}{\text{no. of markers}}$, $y_L^0 = \frac{\sum y\text{-comp of markers}}{\text{no. of markers}}$. Since the approximated Heaviside functions can be grouped into two: big or small support in the interval $[-\epsilon, \epsilon]$, see Chapter 3, we consider one for each group here.

$$H_{1\epsilon}(z) = \begin{cases} 0 & z < -\epsilon \\ \frac{1}{2}\left[1 + \frac{z}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi z}{\epsilon})\right] & |z| \leq \epsilon \\ 1 & z > \epsilon \end{cases}, \quad \delta_{1\epsilon}(z) = \begin{cases} 0 & z < -\epsilon \\ \frac{1}{2}\left[\frac{1}{\epsilon} + \frac{1}{\epsilon}\cos(\frac{\pi z}{\epsilon})\right] & |z| \leq \epsilon \\ 1 & z > \epsilon \end{cases},$$

$$H_{2\epsilon}(z) = \frac{1}{2}(1 + \frac{2}{\pi}\arctan(\frac{z}{\epsilon})), \qquad \delta_{2\epsilon}(z) = \frac{1}{\epsilon\pi(1 + x^2/\epsilon^2)},$$

As explained in §2.2.3, $H_{2\epsilon}$ has a bigger support in the interval $[-\epsilon, \epsilon]$, which means that a moderately large ϵ may lead to spurious results.

In test comparisons, the initial local level set initialization and the choices of parameters are the same for the Badshah-Chen method [12].

Note: As far as selective segmentation is concerned, easier problems refer to those images where the selective target is well separated from all other nearby features; in the extreme case where the separation distance is extremely large and the target feature is of a simple convex shape, one may even use the non-selective models such as [39, 92] by starting evolving contours near the geometric markers.

5.5.1 Test Set 1 — robustness of the new model

First we show some easier problems referring to those images where the selective target is well separated from all other nearby features. Numerical results of our new method for segmenting 8 different images will be shown.

The top left image in Figure 5.4 shows an image with many features where the spiral was the aim of detection. The top right image shows results of all features captured by our global level set, and the last images show the segmentation result of the spiral using 3 geometric markers.

In Figure 5.5, we test the model on a real CT image where the right kidney is to be selected; again the bottom two images show the correctly segmented organ, using 3 geometric markers.

Figure 5.6, shows three test results (of an artificial flower, the cameraman and a cell image) by our model; clearly our selection model delivers good results.

Finally Figure 5.7 shows three more results obtained from segmentation of images with strong noise or smooth contours. Again our model gives the correctly segmented results satisfying the expected selection requirement.

5.5.2 Test Set 2 — comparison of segmentation of easier problems

For easy problems, cases where the separation distance is of more than 3 pixels away and the target feature is of a simple convex shape, one may rely on the Badshah-Chen [12] or Gout-Guyader [67, 69] models by starting evolving contours near the geometric markers. Here we compare our model with Badshah-Chen [12] for three easier problems as shown in Figures 5.8, 5.9, 5.10. Comparative results between [12] and [67, 69] can be found in [12]. For the test results in Figure 5.10, although both models give almost identical segmentation using $H_{1\epsilon}$, the Badshah-Chen method is more sensitive to the ϵ parameter choice for the regularized Heaviside $H_{2\epsilon}$; specifically our model would work for $\epsilon = 0.01$, or $\epsilon = 1$ while the Badshah-Chen method must use the smaller parameter (otherwise redundant features are captured).

5.5.3 Test Set 3 — comparison of segmentation of harder problems

In this set of 4 test problems we consider harder and more challenging cases. In this set the separation distance between features is small or the intensity difference between features and background is small. In these difficult cases, the previous models from [12, 67, 69] will not work. The experimental results from these methods will be shown in each of these examples. Figures 5.11, 5.12, 5.13 and 5.14 show four respective images and their segmented results of one feature; in each case, the top line of images shows the results of [12] which are not correct due to inclusion of redundant features in the selective segmentation and the bottom line shows the correctly segmented results by our new model. Clearly our model is robust.

5.5.4 Test Set 4 — necessity of a selection model

Here we show one final experiment of a selective segmentation model in clear contrast to other widely known methods for global segmentation.

In Figure 5.15, we compare three sets of usual segmentation results with our selective segmentation result. Here the image in Figure 5.15 (a) is the original image, given with the markers indicating where the feature is to be extracted. First (a) is segmented by

the Chan-Vese [39] algorithm to obtain the segmented image in Figure 5.15 (b). Then two cropped and smaller images (c)-(d) of Figure 5.15 (a) are respectively segmented to give the results in (e)-(f). Finally our proposed method gives the correctly segmented result in Figure 5.15 (g)-(h).

Clearly one observes the correct segmentation in such situations where selection is required and which only can be delivered by a selective model such as ours.

5.6 Conclusions

Selective image segmentation is an important and practical problem in image processing, where only certain image features defined by geometric constraints are desired. In this chapter we presented a new variational model with two level set functions (one for global segmentation and the other for local and selective segmentation) for reliable segmentation, improving on two related models proposed recently. An efficient AOS algorithm for solving and helping speed up the dual level set model has been developed. Numerical experiments confirmed that the new model delivers similar results for easier problems to old models [12] and equally reliable results for harder problems where old models fail.

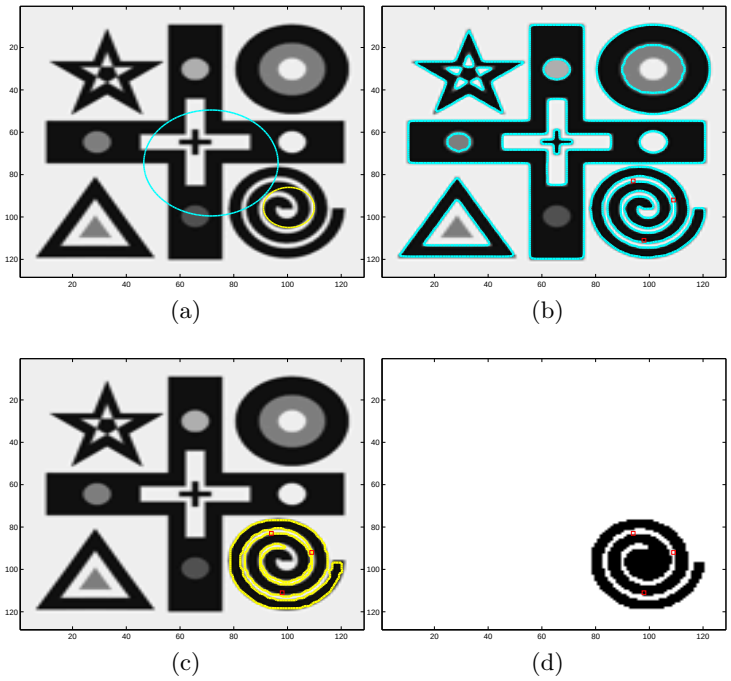


Figure 5.4: Problem 1 of Test Set 1 by the new model — Successful detection of the spiral in a clean and synthetic image with 3 markers. (a) Initial zero level set contours with $dt = 0.1$ (n=128); (b) Successful global segmentation by the New model; (c-d) Successful local segmentation and the segmented feature. CPU time= 56.3 seconds.

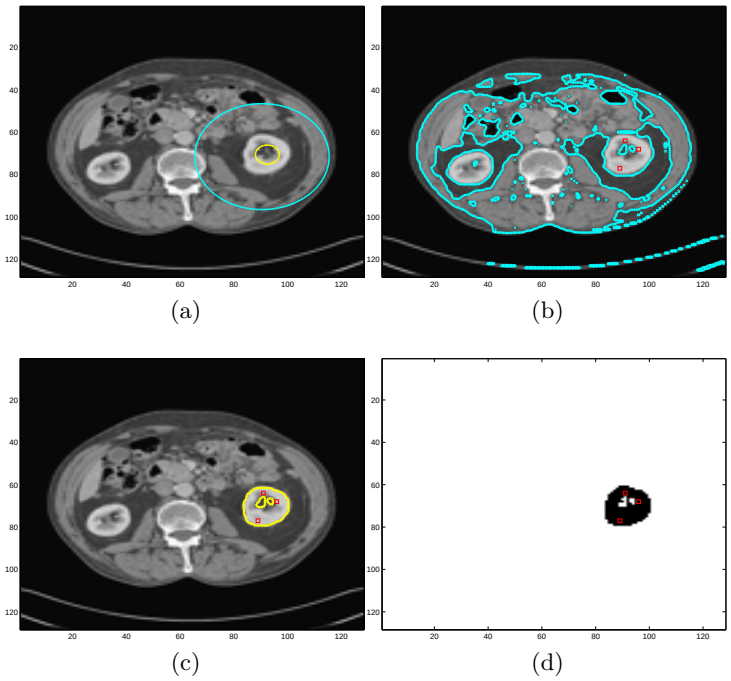


Figure 5.5: Problem 2 of Test Set 1 by the new model — Successful detection of the right kidney in a real CT image with 3 markers. (a) Initial zero level set contours with $dt = 0.01$ (n=128); (b) Successful global segmentation by the New model; (c-d) Successful local segmentation and the segmented feature. CPU time= 69.1 seconds.

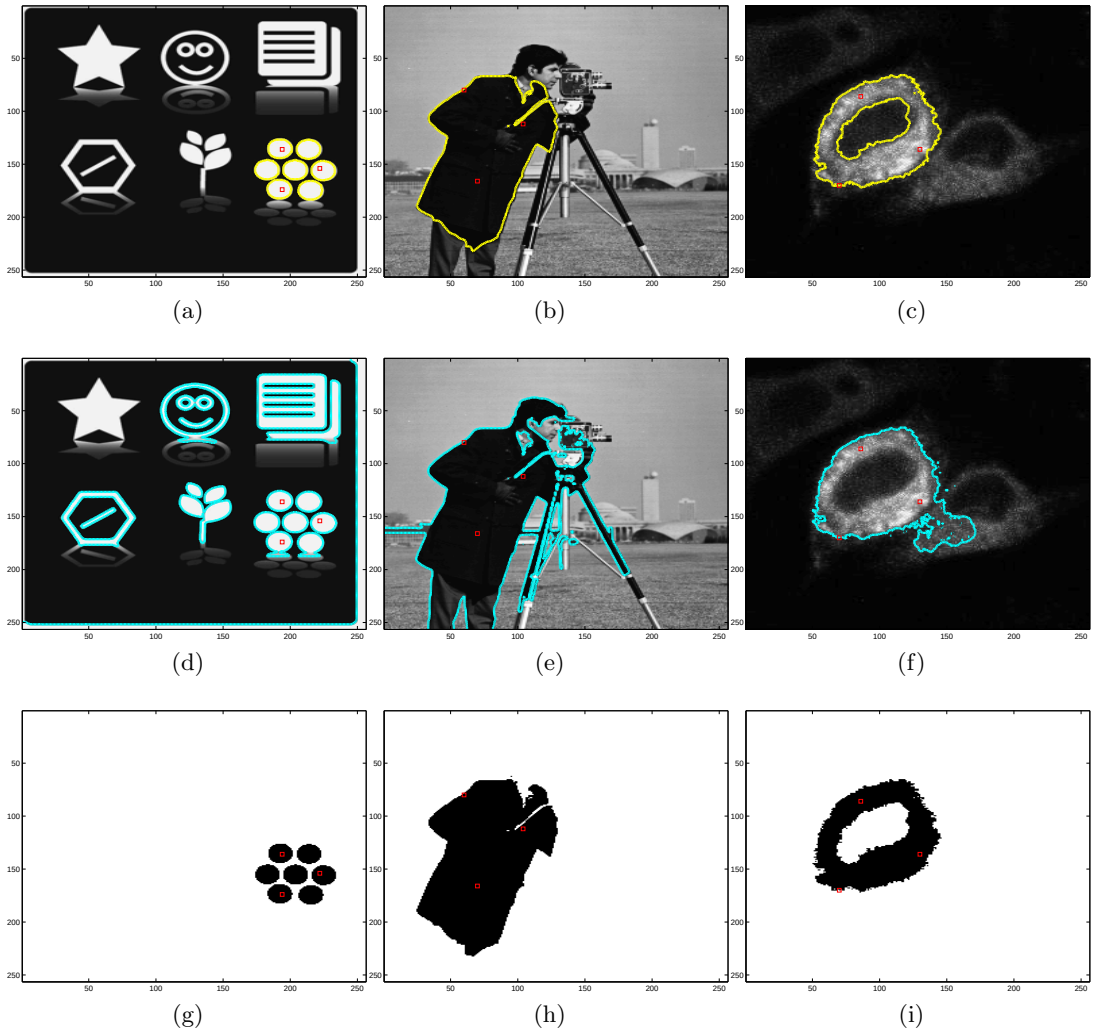


Figure 5.6: Problems 3-5 of Test Set 1 by the new model — (a) Successful detection of the flower in a clean and synthetic image with 3 markers; (b) cameraman in a clean and real image with 3 markers; (c) one cell in a real image with 3 markers. Here we take $dt = 0.1$ (n=256). The first row shows the selected object using 3 markers, the second row the final global level set selection and the third row the selected feature. CPU time 120.4, 138.2 and 517.8 seconds respectively.

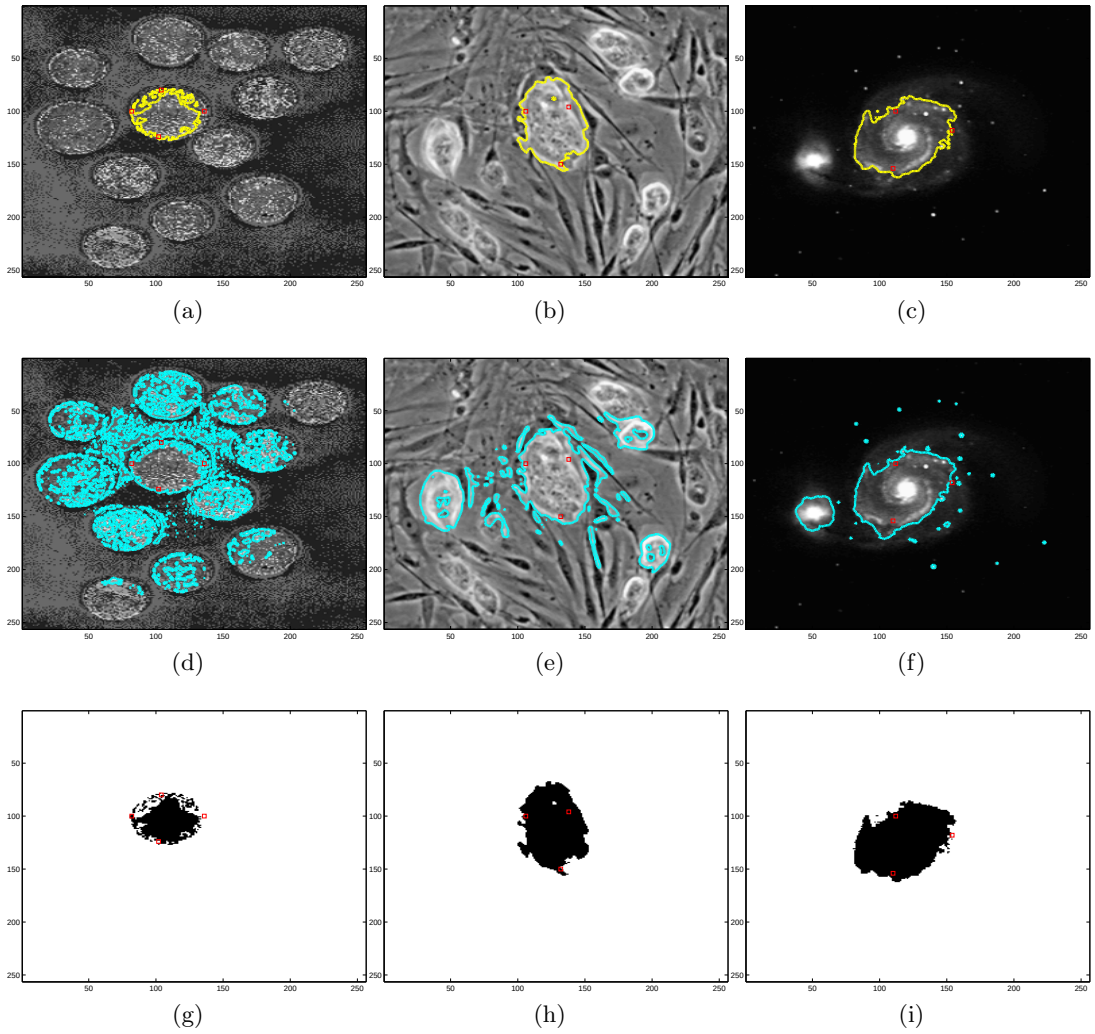


Figure 5.7: Problems 6-8 of Test Set 1 by the new model — (a) Successful detection of one coin in a strong noise image with 3 markers; (b) one cell in a real image of mouse embryonic stem cells with 3 markers; (c) selection of the main galaxy with 3 markers. Here we take $dt = 0.01$ (n=256). The first row shows the selected object from using 3 markers, the second row the final global level set selection and the third row the selected feature. CPU time 221.2, 885.0 and 183.3 seconds respectively.

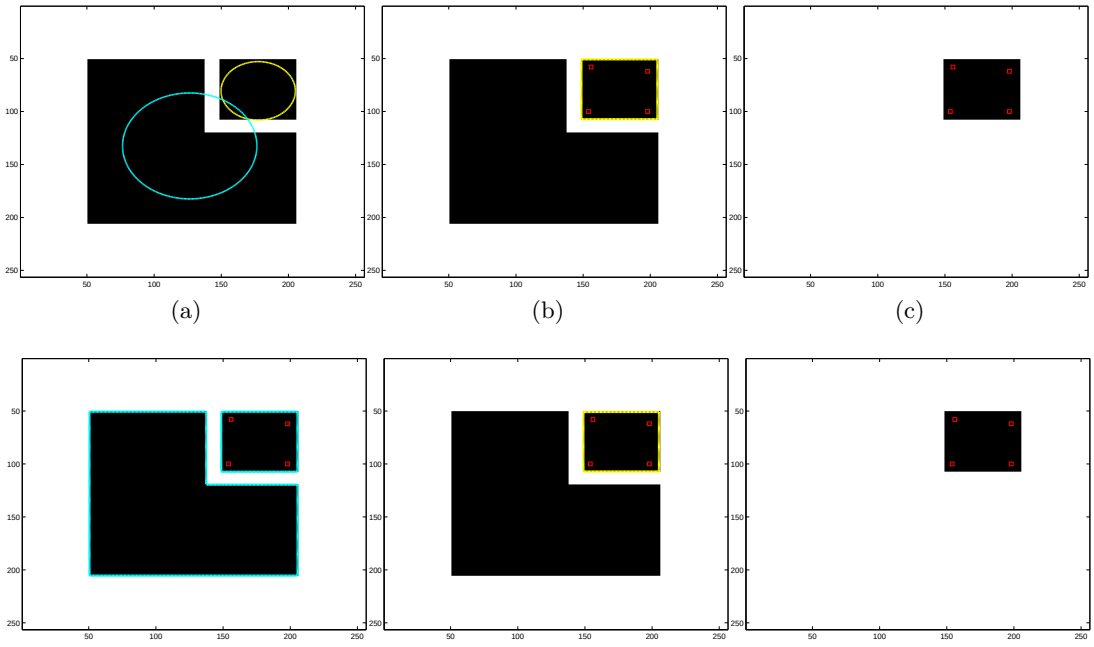


Figure 5.8: Problem 1 in Test Set 2: Identical results by [12] (top row) and this model (bottom row). (a) Initial zero level set contours with $dt = 0.01$ (n=256); (b-c) Successful result and the selected feature with [12], with CPU time= 48.5 seconds; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of the box with 4 markers by the New model, with CPU time= 117.6 seconds.

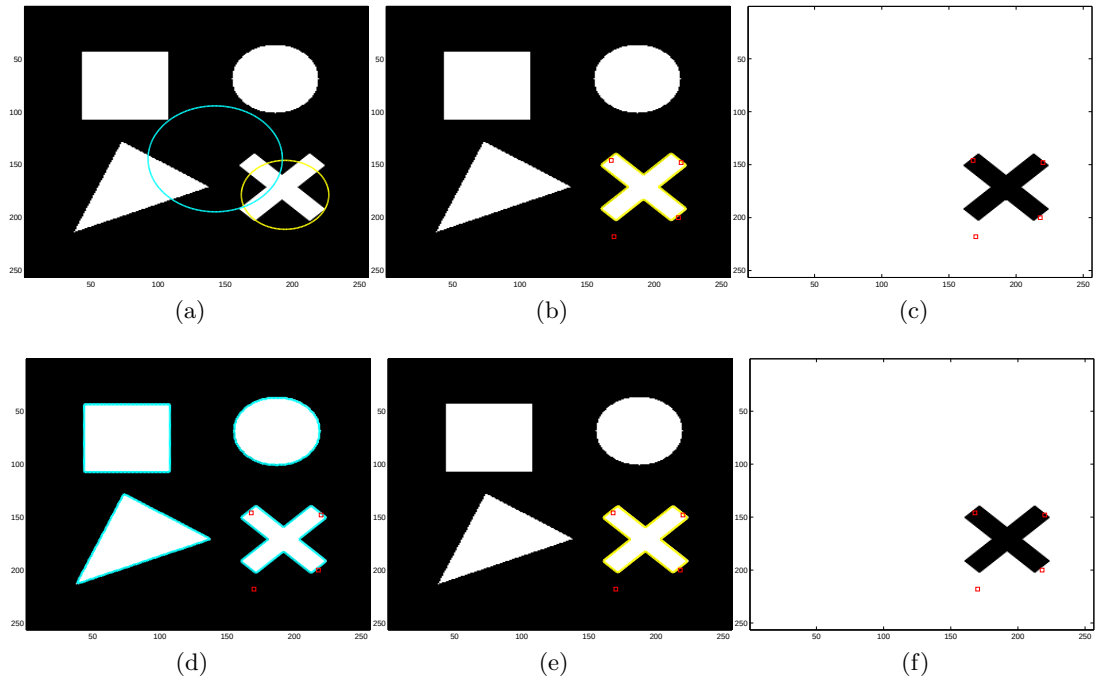


Figure 5.9: Problem 2 in Test Set 2: Identical results by [12] (top row) and this model (bottom row). (a) Initial zero level set contours with $dt = 0.1$ (n=256); (b-c) Successful result and the selected feature with [12], with CPU time= 50.7 seconds; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of the cross with 4 markers by the New model, with CPU time= 118.8 seconds.

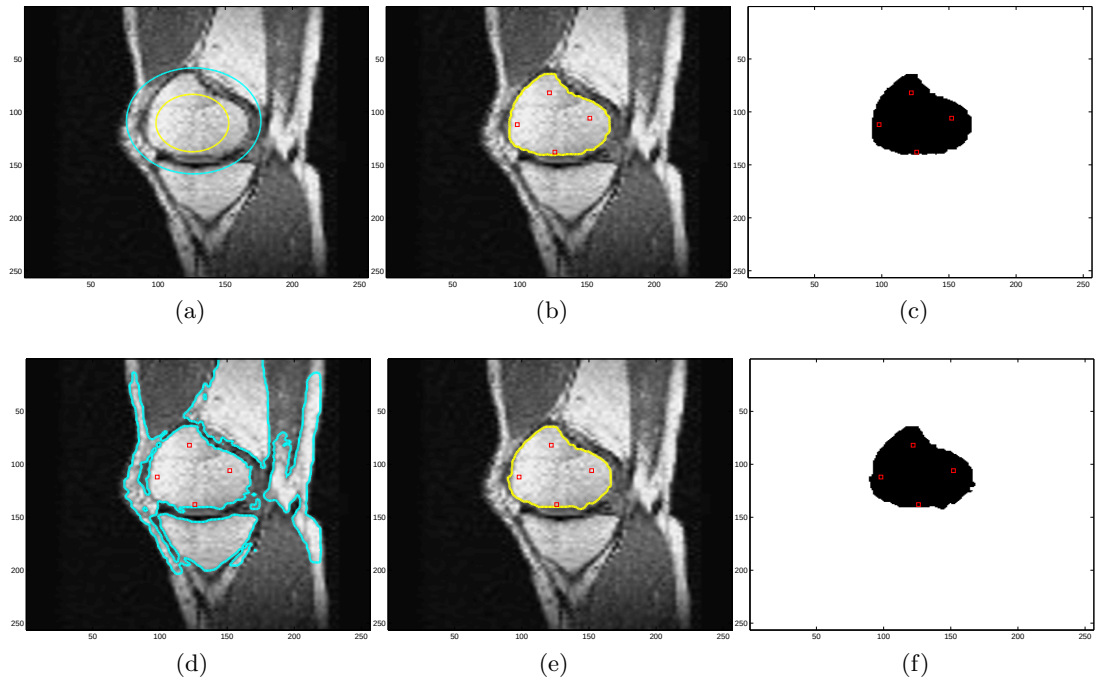


Figure 5.10: Problem 3 in Test Set 2: Identical results by [12] (top row) and this model (bottom row). (a) Initial zero level set contours with $dt = 0.1$ (n=256); (b-c) Successful result and the selected feature with [12], with CPU time= 62.7 seconds; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of the knee cap with 4 markers by the New model, with CPU time= 142.3 seconds.

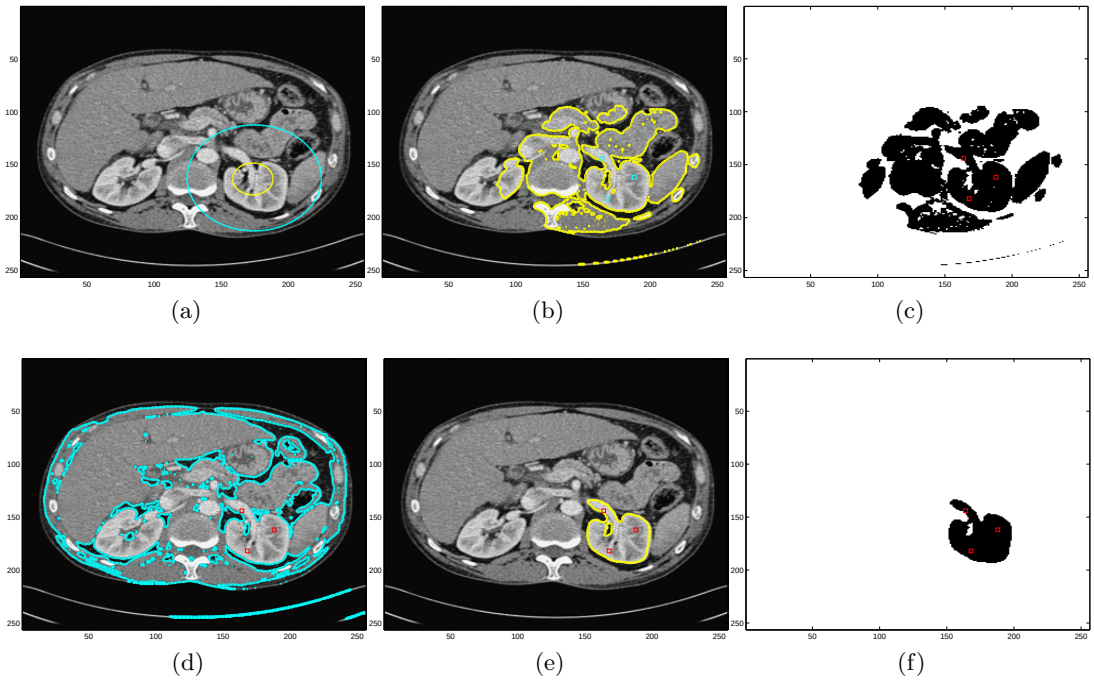


Figure 5.11: Test Set 3 – Comparative results for Problem 1: (a) Initial zero level set contours with $dt = 0.1$ (n=256); (b) Unsuccessful result by [12] model; (c) Redundant features selected with [12]; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of the right kidney with 3 markers with the New model, CPU time= 173.1 seconds.

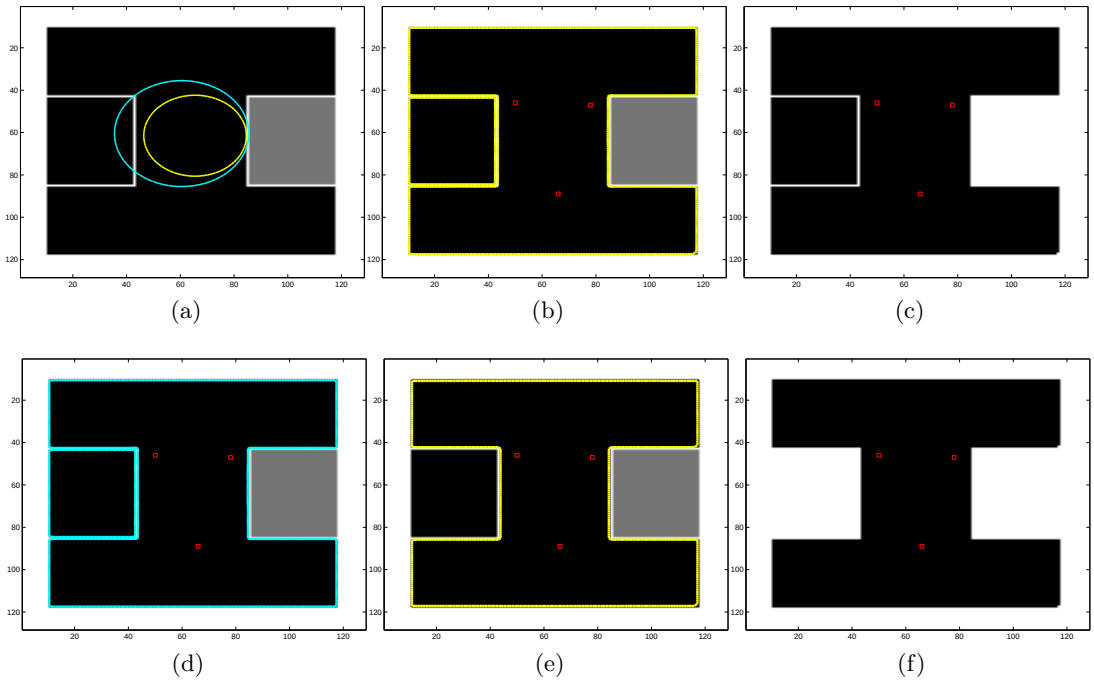


Figure 5.12: Test Set 3 – Comparative results for Problem 2: (a) Initial zero level set contours with $dt = 0.01$ (n=128); (b) Unsuccessful result by [12] model; (c) Redundant shapes selected with [12]; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of a non-convex shape with 3 markers with the New model, CPU time= 49.2 seconds.

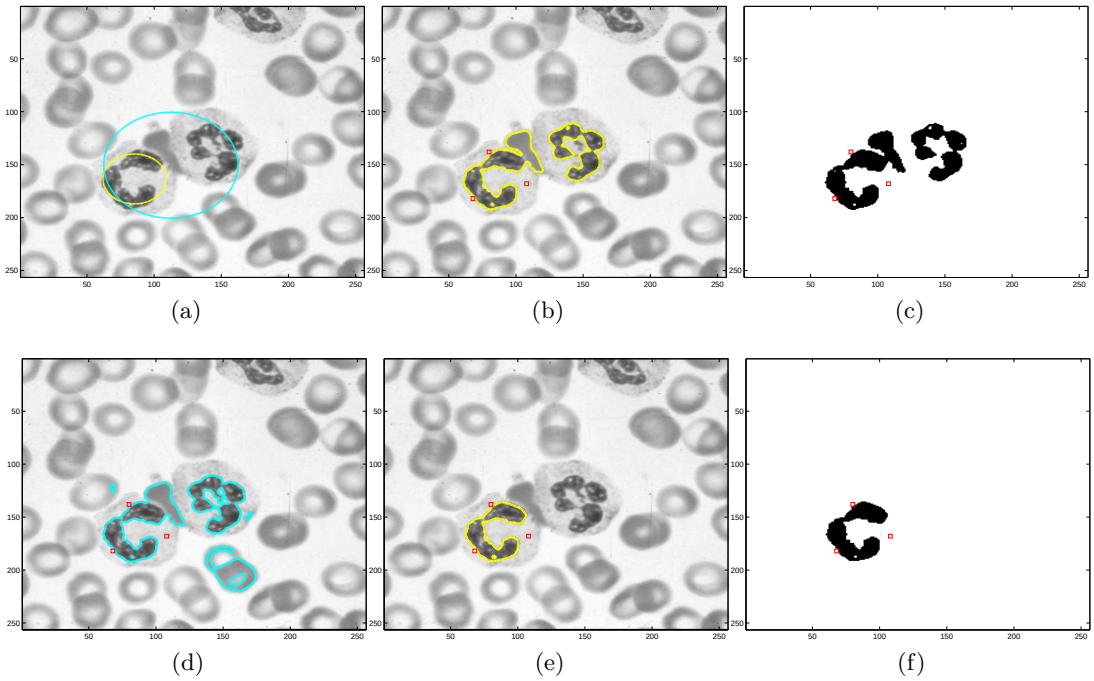


Figure 5.13: Test Set 3 – Comparative results for Problem 3: (a) Initial zero level set contours with $dt = 0.01$ (n=256); (b) Unsuccessful result by [12] model; (c) Redundant cells selected with [12]; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of a single cell with 3 markers with the New model, with CPU time= 217.1 seconds.

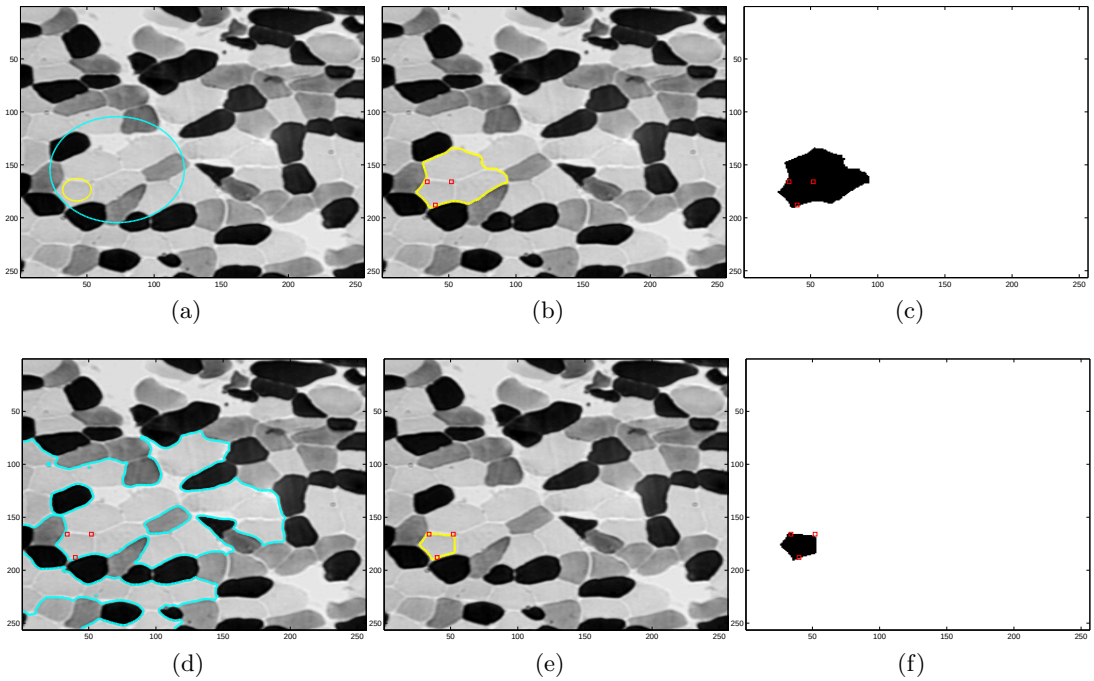


Figure 5.14: Test Set 3 – Comparative results for Problem 4: (a) Initial zero level set contours with $dt = 0.01$ (n=256); (b) Unsuccessful result by [12] model; (c) Redundant cells selected with [12]; (d) Successful global segmentation by the New model; (e-f) Successful local segmentation of a cell with 3 markers with the New model, with CPU time= 178.1 seconds.

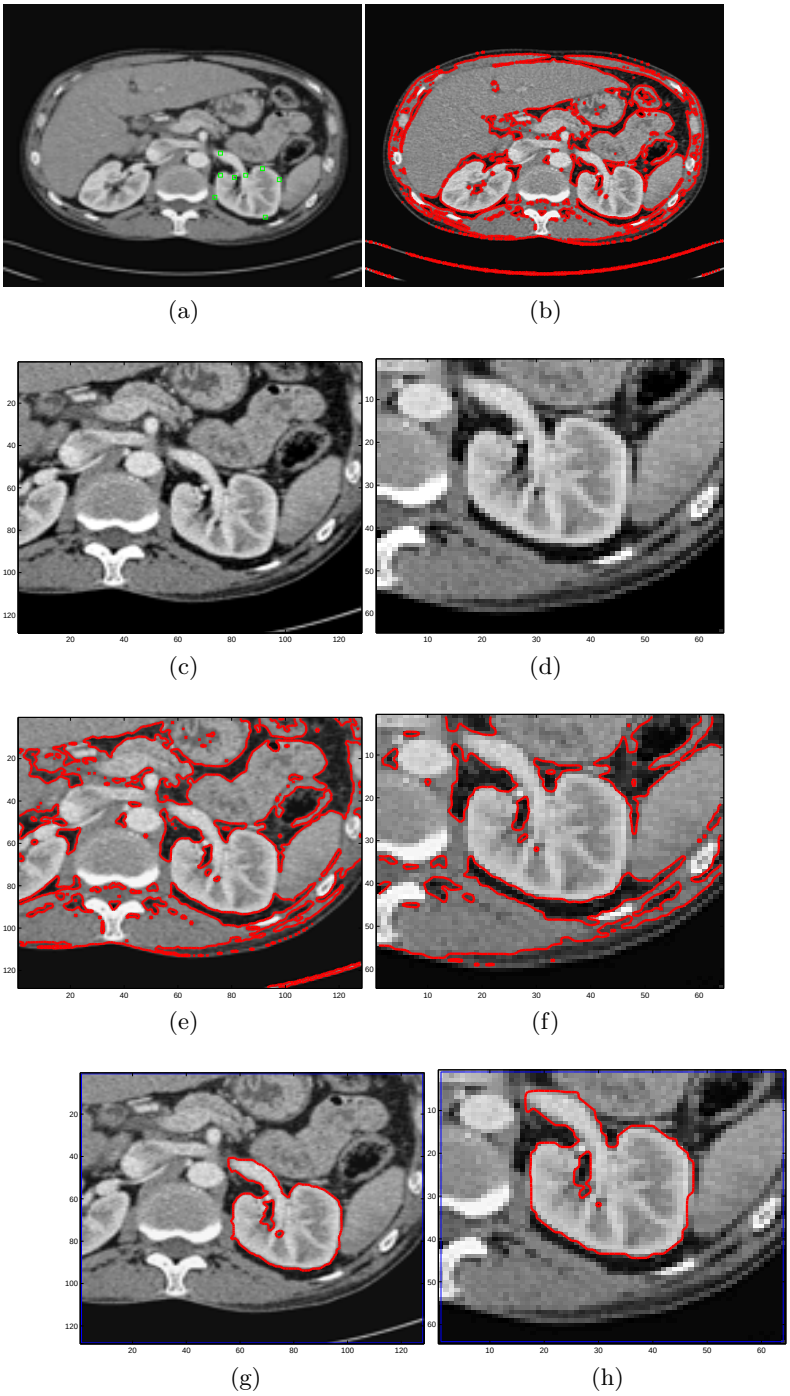


Figure 5.15: Test Set 4 – Comparative results for Problem 4: (a) Given image with the markers set (n=256); (b) Result by [39] model; (c-d) Cropped image size 128×128 and 64×64; (e-f) Segmentation by [39] model; (g-h) Successful segmentation of the object with the New model.

Chapter 6

A Three-dimensional Variational Model for Local Segmentation

This chapter presents a dual level set selective segmentation model that is capable of automatically capturing a local feature/object of interest in three dimensions. This will be an extension of the 2-D model already discussed in Chapter 5. Numerical tests show that the proposed model is robust in locally segmenting complex image structures.

6.1 Introduction

Various models, algorithms and techniques in 2-D may be generalized to the 3-D case, e.g. methods based on 2-D level set active contours may be generalized to 3-D level set active surfaces [40, 176, 180]. These methods have been widely used and successful for many applications where all features/objects in a given image have to be segmented. However, as shown in Chapter 5 there are some other applications where the selection of one feature/object among many is required. This kind of problem leads to a new and challenging task of selective segmentation.

There exist only a few works for selective segmentation in 3-D which are interactive methods or semi-automatic. In a similar way to 2-D selective segmentation these methods are based on active contours [12, 69, 177], which for a proper selection of the initial curve lead to convergence to the aimed object. Since many application fields such as medical imaging, geological surveying and computational fluid dynamics can greatly benefit from 3-D selective segmentation, we find task of the moment to investigate the suitability of generalized 3-D selection models.

In this chapter we will first show a generalized 3-D selective segmentation model based on Badshah-Chen level set technique [12]. We then propose an improved model for a more robust solution of the selective segmentation problem using dual level set selective segmentation. In each case, by creating an initial surface in the target image and evolve regions of interest to reach the intended target object.

The rest of the chapter is organized as follows. We shall first start with a review of the Chan-Vese image segmentation model in 3-D in Section §6.2. Then in Section §6.3 we extend the 2-D selective segmentation Badshah-Chen model [12] into 3-D and develop an AOS method for solving the underlying PDE. Some successful numerical results and comparison with Gout-Guyader model [69] are shown to indicate that this method can be useful. In Section §6.4 we develop a 3-D dual level set selective segmentation model and describe an AOS algorithm for subsequent solutions. Section §6.5

shows further experimental results to demonstrate the advantages of the dual level set selective segmentation model over the generalized Badshah-Chen and Gout-Guyader model in 3-D. Both synthetic and real images are used. The last Section §6.6 concludes the chapter.

6.2 Review of the Chan-Vese Image Segmentation Model in 3-D

We now review the Chan-Vese model [37, 39] in the 3-D framework. Given image $Z : \Omega \subset \mathbb{R}^3 \to \mathbb{R}$, let $\Gamma \subset \Omega$ be a 2-D surface internal boundary in Ω that separates Ω into two subregions, features and their background, with $\Omega = inside(\Gamma) \cup \Gamma \cup outside(\Gamma)$. The Chan-Vese model in 3-D finds the desired interface Γ by minimization of the energy functional [39]:

$$F(\Gamma, c_1, c_2) = \mu \int_\Gamma ds + \lambda_1 \int_{\text{inside}(\Gamma)} |Z(\mathbf{x}) - c_1|^2 d\mathbf{x} + \lambda_2 \int_{\text{outside}(\Gamma)} |Z(\mathbf{x}) - c_2|^2 d\mathbf{x}, \quad (6.1)$$

where c_1, c_2 are respectively the (unknown) mean intensities of the two subregions. Representing Γ by the zero level set Lipschitz function $\phi : \Omega \to \mathbb{R}$

$$\begin{cases} \Gamma = \partial\omega = \{(x, y, z) \in \Omega \mid \phi(x, y, z) = 0\}, \\ \text{inside}(\Gamma) = \{(x, y, z) \in \Omega \mid \phi(x, y, z) > 0\}, \\ \text{outside}(\Gamma) = \{(x, y, z) \in \Omega \mid \phi(x, y, z) < 0\}, \end{cases}$$

and denoting the regularized Heaviside function and Delta function [37] respectively as

$$H_\epsilon(x) = \frac{1}{2}\Big(1 + \frac{2}{\pi}\arctan(\frac{x}{\epsilon})\Big), \qquad \delta_\epsilon(x) = H'_\epsilon(x) = \frac{\epsilon}{\pi(\epsilon^2 + x^2)}, \quad (6.2)$$

the minimization of the regularized functional $F_\epsilon(\phi, c_1, c_2)$ of $F(\phi, c_1, c_2)$ is

$$\begin{aligned} F_\epsilon(\phi(\mathbf{x}), c_1, c_2) = \mu \int_\Omega \delta_\epsilon\Big(\phi(\mathbf{x})\Big)|\nabla\phi(\mathbf{x})| d\mathbf{x} + \lambda_1 \int_\Omega |Z(\mathbf{x}) - c_1|^2 H_\epsilon(\phi) d\mathbf{x} \\ + \quad \lambda_2 \int_\Omega |Z(\mathbf{x}) - c_2|^2 (1 - H_\epsilon(\phi)) d\mathbf{x}, \end{aligned} \quad (6.3)$$

For fixed c_1 and c_2, this yields the following Euler-Lagrange equation for ϕ:

$$\begin{cases} \delta_\epsilon(\phi(\mathbf{x}))\Big[\mu \text{div}\Big(\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\Big) - \lambda_1(Z(\mathbf{x}) - c_1)^2 + \lambda_2(Z(\mathbf{x}) - c_2)^2\Big] = 0 \quad \text{in } \Omega, \\ \frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} = 0 \quad \text{on } \partial\Omega, \end{cases} \quad (6.4)$$

where $\vec{n}$ denotes the unit normal exterior and $\frac{\partial\phi}{\partial\vec{n}}$ is the normal derivative of ϕ at $\partial\Omega$. The segmented image is generated by

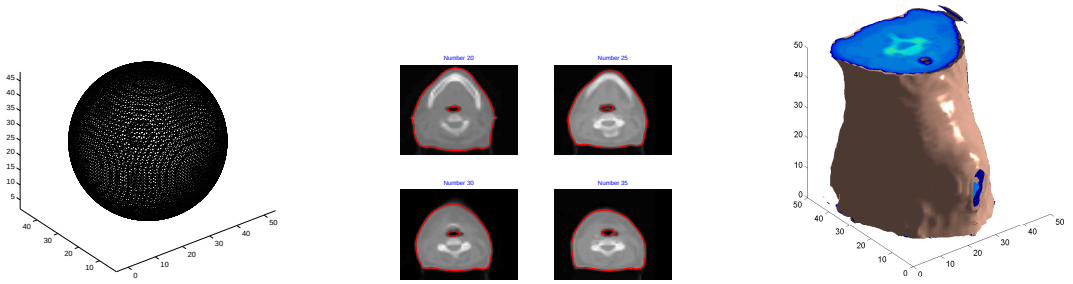
$$u(\mathbf{x}) = H(\phi(\mathbf{x}))c_1 + (1 - H(\phi(\mathbf{x})))c_2.$$

A time marching method that can be used to solve (6.4) brings us to the solution of the following parabolic equation

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon(\phi)\Big[\mu \,\text{div}\,\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big) - \lambda_1(Z(\mathbf{x}) - c_1)^2 + \lambda_2(Z(\mathbf{x}) - c_2)^2\Big], \quad (6.5)$$

with Neumann boundary conditions. For a sufficiently small time step Δt, one attempts to compute $\lim_{t\to\infty} \phi(x, y, z, t) = \phi(x, y, z)$. To speed up the convergence fast methods can be developed such as the AOS method [171, 106, 175], or a multigrid method [160, 180]. Figure 6.1 and 6.2 show a successfully segmented image with the Chan-Vese model in 3-D.

Although one expects the CV model to be able to extract one object, such as one sphere in Figure 6.2, given a local initialization, in fact it can fail to do so in this example because all spheres are close to each other and have the same intensity c_1.

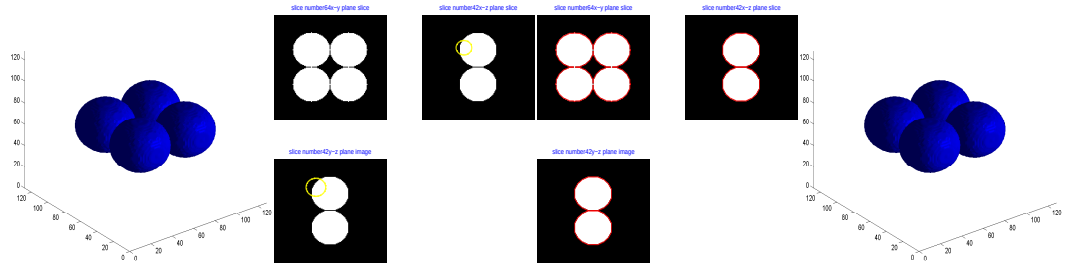


(a) The initial level set

(b) Several slices of the segmented object

(c) Level set after the processing

Figure 6.1: Result of segmenting a 3D CT brain data set in the resolution of $64 \times 64 \times 64$ with $dt = 0.1$, $\lambda_1 = \lambda_2 = 1$,$\epsilon = 1$, and 40 iterations.



(a) Given 3-D volumetric objects

(b) in yellow a 2-D projection of the first initial-view from xy, yz and zx plan; segmented object

(c) segmented object

(d) The constructed 3-D Object after segmentation

Figure 6.2: Results of segmenting an artificial geometrical image of resolution $128 \times 128 \times 128$ with 4 spheres. The time marching time step is $dt = 0.01$ with $\lambda_1 = \lambda_2 = 1$ in 100 iterations.

6.3 A Generalized Badshah-Chen Model in 3-D

Similar to how the 3-D Chan-Vese model is generalized from its original 2-D formulation, we can generalize the Badshah-Chen 2-D selective segmentation method [12] to selective segmentation in 3-D and assess its effectiveness. We have to remind that in comparison with Gout-Guyader [69] the Badshah-Chen model [12] has the presence of a region fitting term added by Badshah-Chen.

Let us define a cuboid domain Ω and some given geometrical points in a set $\mathcal{A} = \{w_i^* = (x_i^*, y_i^*, z_i^*) \in \Omega, \quad 1 \leq i \leq n_1\} \subset \Omega$ consisting of n_1 distinct points near the

object Γ [67, 69, 12]. In a selective segmentation, we hope to detect the features $Z(x, y, z)$ that are defined in a closed domain which is the closest to $\mathcal{A}$.

6.3.1 Badshah-Chen Time Marching Model in 3-D

The edge detector $g(Z(\mathbf{x}))$ and the distance function $d(\mathbf{x})$ are two quantities used in the Badshah-Chen [12] model with the property that they approach zero when $\mathbf{x} = (x, y, z)$ is near Γ, and is large when away from it. Similar to the edge detector function g used in 2-D models [12, 28, 67], we can define the 3-D version as

$$g(Z(\mathbf{x})) = \frac{1}{1 + |\nabla G_\sigma(\mathbf{x}) * Z(\mathbf{x})|^2}, \tag{6.6}$$

where $G_\sigma(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-|\mathbf{x}^2|/2\sigma^2}$ is a Gaussian kernel function with $\sigma > 0$ and the localization property that $G_\sigma(\mathbf{Z})$ decreases and approaches zero as $|\nabla \mathbf{Z}|$ increases. G_σ is needed when the given image $\mathbf{Z}$ has some noise (usually we take in case of noise $\sigma = 1/2$); otherwise it is not needed. Similar to 2-D we define the 3-D distance function d in the following way:

$$d(\mathbf{x}) = \text{distance}(\mathbf{x}, \mathcal{A}) = \prod_{i=1}^{n_1} \left(1 - e^{-\frac{(\mathbf{x} - \mathbf{x}_i^*)^2}{2\tau^2}}\right), \qquad \forall (\mathbf{x}) \in \Omega, \tag{6.7}$$

where τ is a positive constant. Clearly d acts locally and will be approximately 0 in the neighborhood of points of $\mathcal{A}$. The aim is to find a surface Γ such that $d \simeq 0$ or $g \simeq 0$ along it.

The generalized Badshah-Chen selective segmentation can be written in the following way:

$$\begin{aligned} \min_{\Gamma, c_1, c_2} F(\Gamma, c_1, c_2) = &\int_\Gamma d(\mathbf{x}) g(|\nabla Z(\mathbf{x})|) ds + \\ &\lambda_1 \int_{inside(\Gamma)} |Z(\mathbf{x}) - c_1|) d\mathbf{x} + \lambda_2 \int_{outside(\Gamma)} |Z(\mathbf{x}) - c_2|) d\mathbf{x} \end{aligned} \tag{6.8}$$

The surface Γ will stop at a local minimum where $d \simeq 0$ (in the neighborhood of points for $\mathcal{A}$) or $g \simeq 0$ (near object boundaries).

Introducing the level set and the Heaviside and Dirac delta function the proposed model can be rewritten:

$$\begin{aligned} \min_{\phi(\mathbf{x}), c_1, c_2} F(\phi(\mathbf{x}), c_1, c_2) = &\mu \int_\Omega d(\mathbf{x}) g(|\nabla Z(\mathbf{x})|) |\nabla H(\phi(\mathbf{x}))| d\mathbf{x} \\ &+ \lambda_1 \int_\Omega |Z(\mathbf{x}) - c_1|^2 H(\phi(\mathbf{x})) d\mathbf{x} \\ &+ \lambda_2 \int_\Omega |Z(\mathbf{x}) - c_2|^2 (1 - H(\phi(\mathbf{x}))) d\mathbf{x}, \end{aligned} \tag{6.9}$$

To avoid the non-differentiability at the origin we use the regularized H_ϵ as before and

the minimization problem is

$$\min_{\phi(\mathbf{x}),c_1,c_2} F_\epsilon(\phi(\mathbf{x}),c_1,c_2) = \mu\int_\Omega W\delta_\epsilon(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|d\mathbf{x} + \lambda_1\int_\Omega |Z(\mathbf{x})-c_1|^2 H_\epsilon(\phi(\mathbf{x}))d\mathbf{x} + \lambda_2\int_\Omega |Z(\mathbf{x})-c_2|^2(1-H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}, \quad (6.10)$$

where $W = d(\mathbf{x})g(|\nabla Z(\mathbf{x})|)$. Keeping c_1 and c_2 fixed, one can minimize (6.10) with respect to $\phi(\mathbf{x})$ and derive the following Euler-Lagrange equation for ϕ

$$\delta_\epsilon(\phi)\mu\nabla\cdot\left(W\frac{\nabla\phi}{|\nabla\phi|}\right) + \delta_\epsilon(\phi)\left(-\lambda_1(Z(\mathbf{x})-c_1)^2+\lambda_2(Z(\mathbf{x})-c_2)^2\right) = 0, \quad \text{in } \Omega \quad (6.11)$$

$$\text{with } W\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} = 0, \quad \text{on } \partial\Omega,$$

with c_1 and c_2 in the same way as in the Chan-Vese model

$$c_1(\phi(\mathbf{x})) = \frac{\int_\Omega Z(\mathbf{x})H_\epsilon(\phi(\mathbf{x}))d\mathbf{x}}{\int_\Omega H_\epsilon(\phi(\mathbf{x}))d\mathbf{x}} \quad (6.12)$$

$$c_2(\phi(\mathbf{x})) = \frac{\int_\Omega Z(\mathbf{x})(1-H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}}{\int_\Omega(1-H_\epsilon(\phi(\mathbf{x})))d\mathbf{x}}. \quad (6.13)$$

A balloon term can be added in (6.11) to speed up the convergence

$$\delta_\epsilon(\phi)\mu\nabla\cdot\left(W\frac{\nabla\phi}{|\nabla\phi|}\right) + \delta_\epsilon(\phi)\left(-\lambda_1(Z(\mathbf{x})-c_1)^2+\lambda_2(Z(\mathbf{x})-c_2)^2\right) + \alpha|\nabla\phi|W = 0, \quad \text{in } \Omega, \quad (6.14)$$

with $\frac{\partial\phi}{\partial\vec{n}} = 0$, on $\partial\Omega$.

6.3.2 An Additive Operator Splitting Algorithm for Badshah-Chen Model

Since we are dealing with 3-D data, one needs a fast iterative algorithm to solve (6.14). Once the nonlinear coefficients in (6.14) are frozen in a fixed-point fashion, an additive operator splitting algorithm similar to [12, 69, 171, 175, 106] can be developed. Denoting $F = \dfrac{W}{|\nabla\phi|}$ in equation (6.14) we have

$$\frac{\partial\phi}{\partial t} = \mu\delta_\epsilon(\phi)\nabla\cdot(F\nabla\phi) + f, \quad (6.15)$$

where $f = \delta_\epsilon(\phi)(-\lambda_1(u-c_1)^2+\lambda_2(u-c_2)^2)+\alpha|\nabla\phi|W$. The advantage of the AOS is the way how the method can split the m-dimensional spatial operator into a sum of m one-dimensional space discretization. Here we consider the following three dimensional problem to be solved

$$\frac{\partial\phi}{\partial t} = \mu\delta_\epsilon(\phi)(\partial_x(F\partial_x\phi)+\partial_y(F\partial_y\phi)+\partial_z(F\partial_z\phi))+f, \quad (6.16)$$

which can be rewritten

$$\phi^{n+1} = \phi^n + \Delta t\Big(\mu\delta_\epsilon(\phi^n)A_1(\phi^n)\phi^{n+1} + A_2(\phi^n)\phi^{n+1} + A_3(\phi^n)\phi^{n+1}\Big) + \Delta t f(\mathbf{x})$$

i.e.,

$$3\phi^{n+1} = 3\phi^n + 3\Delta t\Big(\mu\delta_\epsilon(\phi^n)(A_1(\phi^{n+1}) + A_2(\phi^{n+1}) + A_3(\phi^{n+1}))\Big) + 3\Delta t f(\mathbf{x}).$$

Further splitting in x, y and z directions, we have

$$\begin{cases} \phi^{n+1} = \phi^n + 3\Delta t\Big(\mu\delta_\epsilon(\phi^n)A_1(\phi^n)\phi^{n+1}\Big) + \Delta t f(\mathbf{x}), \\ \phi^{n+1} = \phi^n + 3\Delta t\Big(\mu\delta_\epsilon(\phi^n)A_2(\phi^n)\phi^{n+1}\Big) + \Delta t f(\mathbf{x}), \\ \phi^{n+1} = \phi^n + 3\Delta t\Big(\mu\delta_\epsilon(\phi^n)A_3(\phi^n)\phi^{n+1}\Big) + \Delta t f(\mathbf{x}). \end{cases}$$

The AOS solution is

$$\phi^{n+1} = \frac{1}{3}\sum_{l=1}^{3}\phi_l^{n+1}$$

after solving 3 separate equations

$$(\ I - 3\Delta t\mu\delta_\epsilon A_l(\phi^n))\phi_l^{n+1} = \phi^n + \Delta t f(\mathbf{x}) \text{ for } l = 1, 2, 3.$$

In the system above, I is the identity matrix and A_l for $l = 1, 2, 3$ are tridiagonal matrices derived in the following form:

$$(A_1(\phi^n)\phi^{n+1})_{i,j,k} = \left(\partial_x(W^n\frac{\partial_x\phi^{n+1}}{|\nabla\phi^n|})\right)_{i,j,k} = \left(\partial_x(\frac{W^n}{|\nabla\phi^n|}\partial_x\phi^{n+1})\right)_{i,j,k} \overset{CD}{=}$$
$$\frac{F^n_{i+1/2,j,k}(\partial_x\phi^{n+1})_{i+1/2,j,k} - F^n_{i-1/2,j,k}(\partial_x\phi^{n+1})_{i-1/2,j,k}}{h_x} \overset{CD}{=}$$
$$\frac{(\frac{W^n}{|\nabla\phi^n|})_{i+1/2,j,k}(\frac{\phi^{n+1}_{i+1,j,k}-\phi^{n+1}_{i,j,k}}{h_x}) - (\frac{W^n}{|\nabla\phi^n|})_{i-1/2,j,k}(\frac{\phi^{n+1}_{i,j,k}-\phi^{n+1}_{i-1,j,k}}{h_x})}{h_x} =$$
$$A_{1,i,j,k}(\phi^{n+1}_{i+1,j,k} - \phi^{n+1}_{i,j,k}) - A_{2,ij,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i-1,j,k}),$$

$$(A_2(\phi^n)\phi^{n+1})_{i,j,k} = \left(\partial_y(W^n\frac{\partial_y\phi^{n+1}}{|\nabla\phi^n|})\right)_{i,j,k} = \left(\partial_y(\frac{W^n}{|\nabla\phi^n|}\partial_y\phi^{n+1})\right)_{i,j,k} \overset{CD}{=}$$
$$\frac{F^n_{i,j+1/2,k}(\partial_y\phi^{n+1})_{i,j+1/2,k} - F^n_{i,j-1/2,k}(\partial_y\phi^{n+1})_{i,j-1/2,k}}{h_y} \overset{CD}{=}$$
$$\frac{(\frac{W^n}{|\nabla\phi^n|})_{i,j+1/2,k}(\frac{\phi^{n+1}_{i,j+1,k}-\phi^{n+1}_{i,j,k}}{h_y}) - (\frac{W^n}{|\nabla\phi^n|})_{i,j-1/2,k}(\frac{\phi^{n+1}_{i,j,k}-\phi^{n+1}_{i,j-1,k}}{h_y})}{h_y} =$$
$$B_{1,i,j,k}(\phi^{n+1}_{i,j+1,k} - \phi^{n+1}_{i,j,k}) - B_{2,i,j,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i,j-1,k}),$$

$$(A_3(\phi^n)\phi^{n+1})_{i,j,k} = \left(\partial_z(W^n\frac{\partial_z\phi^{n+1}}{|\nabla\phi^n|})\right)_{i,j,k} = \left(\partial_z(\frac{W^n}{|\nabla\phi^n|}\partial_z\phi^{n+1})\right)_{i,j,k} \overset{CD}{=}$$
$$\frac{F^k_{i,j,k+1/2}(\partial_z\phi^{n+1})_{i,j,k+1/2} - F^k_{i,j-1/2}(\partial_z\phi^{n+1})_{i,j-1/2,k}}{h_z} \overset{CD}{=}$$
$$\frac{(\frac{W^n}{|\nabla\phi^n|})_{i,j,k+1/2}(\frac{\phi^{n+1}_{i,j,k+1}-\phi^{n+1}_{i,j,k}}{h_z}) - (\frac{W^n}{|\nabla\phi^n|})_{i,j,k-1/2}(\frac{\phi^{n+1}_{i,jk}-\phi^{n+1}_{i,j,k-1}}{h_z})}{h_z} =$$
$$C_{1,i,j,k}(\phi^{n+1}_{i,j,k+1} - \phi^{n+1}_{i,j,k}) - C_{2,i,j,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i,j,k-1})$$

with

$$A_{1,ij} = \frac{W^n_{i,j,k} + W^n_{i+1,j,k}}{2h_x^2}\frac{1}{\sqrt{(\phi^2_x)_{i+1/2,j,k} + (\phi^2_y)_{i+1/2,j,k} + (\phi^2_z)_{i+1/2,j,k} + \beta}},$$
$$A_{2,i,j,k} = \frac{W^n_{i-1,j,k} + W^n_{i,j,k}}{2h_x^2}\frac{1}{\sqrt{(\phi^2_x)_{i-1/2,jk} + (\phi^2_y)_{i-1/2,jk} + (\phi^2_z)_{i-1/2,jk} + \beta}},$$
$$B_{1,i,j,k} = \frac{W^n_{i,j,k} + W^n_{i,j+1,k}}{2h_y^2}\frac{1}{\sqrt{(\phi^2_x)_{i,j+1/2,k} + (\phi^2_y)_{i,j+1/2,k} + (\phi^2_z)_{i,j+1/2,k} + \beta}},$$
$$B_{2,i,j,k} = \frac{W^n_{i,j-1,k} + W^n_{i,j,k}}{2h_y^2}\frac{1}{\sqrt{(\phi^2_x)_{i,j-1/2,k} + (\phi^2_y)_{i,j-1/2,k} + (\phi^2_z)_{i,j-1/2,k} + \beta}},$$
$$C_{1,i,j,k} = \frac{W^n_{i,j,k} + W^n_{i,j,k+1}}{2h_z^2}\frac{1}{\sqrt{(\phi^2_x)_{ij,k+1/2} + (\phi^2_y)_{ij,k+1/2} + (\phi^2_z)_{ij,k+1/2} + \beta}},$$
$$C_{2,i,j,k} = \frac{W^n_{i,j,k-1} + W^n_{i,j,k}}{2h_z^2}\frac{1}{\sqrt{(\phi^2_x)_{ij,k-1/2} + (\phi^2_y)_{ij,k-1/2} + (\phi^2_z)_{ij,k-1/2} + \beta}},$$

where "CD" stands for central differences.

6.3.3 Experimental Results for Badshah-Chen [12] and Gout-Guyader [69] Selective Models.

Next we present 4 examples using the generalized Badshah-Chen [12] selective model and Gout-Guyader [69]. For these experiments we choose images of size $n = 128$ (i.e. $128 \times 128 \times 128$) tested with $\lambda_1 = \lambda_2 = 1$, $\tau = 4$, $h = 1$ and $\mu = n^2/10$. The initial contour is a sphere given by markers set such that the centre and radius of this sphere are $\mathbf{w_0} = (x_0, y_0, z_0) = (\frac{1}{n_1}\sum_{i=1}^{n_1} x_i^*, \frac{1}{n_1}\sum_{i=1}^{n_1} y_i^*, \frac{1}{n_1}\sum_{i=1}^{n_1} z_i^*)$ and $r = \min_i |\mathbf{w}_i^* - \mathbf{w}_0|$ respectively, where $\mathbf{w}_i^* = (x_i^*, y_i^*, z_i^*)$. In our experiments the regularized Heaviside and Dirac delta functions are the same as in equation (6.2) with $\epsilon = 10^{-2}$ (or other alternatives can be considered such as $H_{2\epsilon} = \frac{1}{2}(1 + \text{erf}(\frac{\epsilon}{x}))$ with $\epsilon = 1$). The result shown has been processed with the AOS algorithm as a faster solver compared with time marching since Δt (the time step) is not required to be too small.

As we see from these experiments the Badshah-Chen [12] and Gout-Guyader [69] models work satisfactorily for cases where nearby features are not too close, Figure 6.3, 6.4 and 6.5, while these two models fail for cases where the objects have the same intensity and are near each other like shown in Figure 6.6 and 6.7. For this example as recommended in both papers [12] and [69], more markers near the points the spheres

touch are needed to help the program to stop. This would not be practical for 3-D images. Other examples, such as Figure 6.8 and a few more in Section 6.4, show that the Badshah-Chen [12] and Gout-Guyader [69] models can totally fail.

The Gout-Guyader [69] model is in the absence of the fitting term if we compare with Badshah-Chen [12], a term which helps the old Gout-Guyader [69] model in presence of noise, but can provide misleading information if objects are near and have the same intensities. In other words, with same argument of Sethian [147] the Gout-Guyader [69] model suffers from the modified effect of g, which is profound, while the Badshah-Chen [12] selective model fails due to the influence of the region term similar to Chan-Vese. This fitting term can cause incorrect results if the number of iterations is big enough or the objects have same intensity c_1. This phenomena can be observed in cases of objects near one another, such as Figure 6.6 and 6.7 with an artificial given image consisting of two symmetric spheres with the same intensities. Both Badshah-Chen [12] and Gout-Guyader [69] struggle to keep the contour in the target object in long term iterations. This is due to the strong influence of the g term, non appropriate region fitting term and the absence of more markers to the points nearby the boundaries with the other sphere.

A much harder case is shown in Figure 6.8, where there is no space between the objects and the intensity difference between the nearby objects is small. The result obtained in this case from the Badshah-Chen and Gout-Guyader models is not correct. Other example for which these models fail are shown in Figure 6.15 and 6.16. We will see from the next Section §6.4 that the dual level set selective segmentation model will overcome all these difficulties. In this Section we shall give a correct segmentation by a new model.

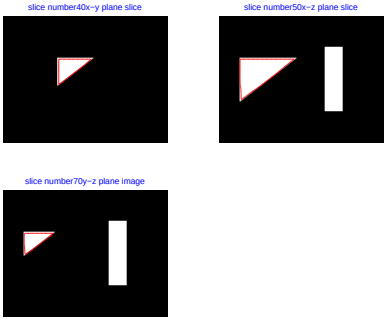
6.4 A 3-D Dual Level Set Variational Model

In this section we propose a dual 3-D level set selective segmentation method, generalizing the new 2-D selective model of Chapter 5, which will allow us to obtain satisfactory results even for some hard cases in which the 3-D Badshah-Chen and Gout-Guyader model fail.

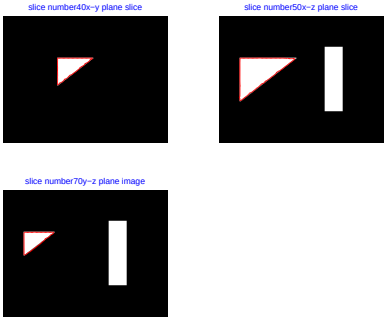
6.4.1 A 3-D Dual Level Set Variational Model

To overcome the problem of failing with Badshah-Chen model (6.11) in cases of similar intensity values to c_1, which implies in this way inclusion of unwanted objects into the selective contour/surface Γ, we consider the new 2-D dual level set approach introduced in Chapter 5. In a similar way, the model in 3-D will hire a global level set ϕ_G to capture all objects with intensities close to c_1 and a local level set ϕ_L to capture the required object.

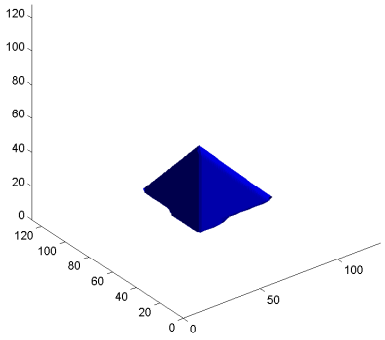
Denote by $\Gamma_G = \partial\Omega_G$ in Ω the global evolving surface for locating all 3-D features Ω_G in image $Z(\mathbf{x})$ and by $\Gamma_L = \partial\Omega_L$ in Ω the desired selective 3-D object, $\Omega_L \subseteq \Omega_G$. Thus we have inside$(\Gamma_L) = \Omega_L$, outside$(\Gamma_L) = \Omega\backslash\overline{\Omega_L}$, inside$(\Gamma_G) = \Omega_G$ and outside$(\Gamma_G) = \Omega\backslash\overline{\Omega_G}$ (see Section §5.3 for more details). Note that the possible advantages of having an enlarged domain of Ω_G within a distance of γ away have not been explored for 3-D. Then the 3-D dual level set selective segmentation has the following



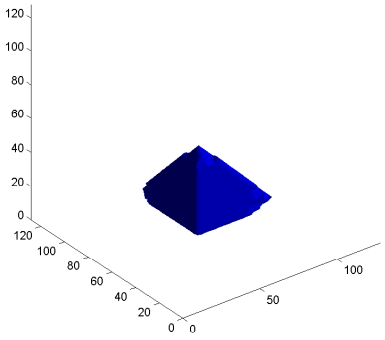
(a) Several slices of the segmented object with Gout-Guyader model [69]



(b) Various slices of the segmented object 3D Badshah-Chen model [12]

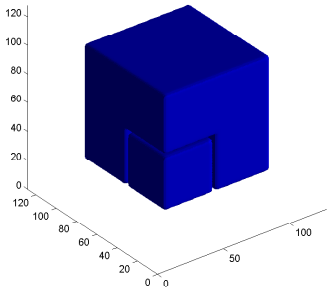


(c) Successful selected object with Gout-Guyader [69] model

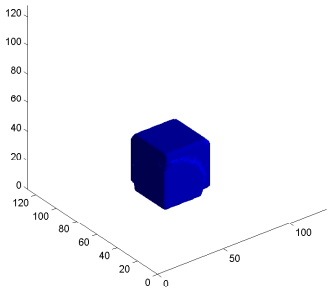


(d) Successful selected object with 3-D Badshah-Chen [12] model

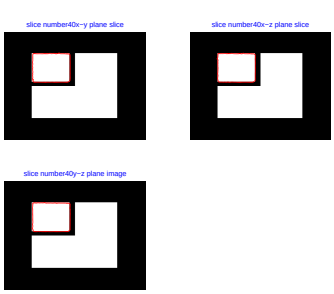
Figure 6.3: Successful segmentation of a geometrical artificial object using the 3-D Badshah-Chen and Gout-Guyader method with an AOS algorithm with $\Delta t = 0.1$.



(a) Tested object

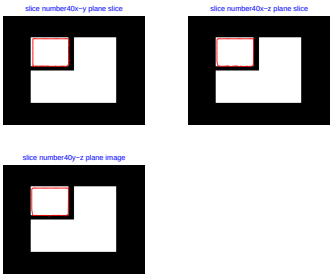


(b) Successful segmentation of the small cuboid with Badshah-Chen model [12]

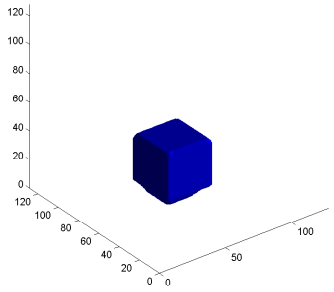


(c) Various slices of the segmented object with Badshah-Chen model [12]

Figure 6.4: Successful segmentation by the Badshah-Chen model [12] of a given geometric volume with two cuboids with a small gap in between with an AOS algorithm with $\Delta t = 0.1$.



(a) Various slices of the segmented object with Gout-Guyader model [69] after 100 iterations



(b) Successful segmentation of the small cuboid with Gout-Guyader model [69] after 100 iterations

Figure 6.5: Successful segmentation by the Gout-Guyader model [69] of a given geometric volume with two cuboids with a small gap in between with an AOS algorithm with $\Delta t = 0.1$.

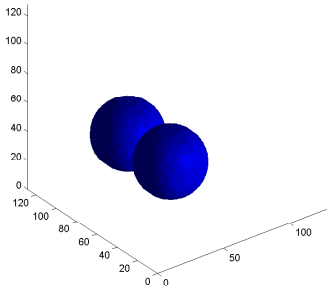
form

$$\begin{aligned}\min_{\phi_L,\phi_G,c_1,c_2} F(\phi_L(\mathbf{x}),\phi_G(\mathbf{x}),c_1,c_2) = \\ \mu_1 \int_\Omega d(\mathbf{x}) g(|\nabla Z(\mathbf{x})|)|\nabla H(\phi_L(\mathbf{x}))|H(\phi_G(\mathbf{x}))d\mathbf{x} + \\ \frac{\mu_{\mathrm{L}}}{2}\int_\Omega (|\nabla\phi_L(\mathbf{x})|-1)^2 d\mathbf{x} + \\ \mu_2 \int_\Omega g(|\nabla Z(\mathbf{x})|)|\nabla H(\phi_G(\mathbf{x}))|d\mathbf{x} + \\ \frac{\mu_{\mathrm{G}}}{2}\int_\Omega (|\nabla\phi_G(\mathbf{x})|-1)^2 d\mathbf{x} + \\ \lambda_{1\mathrm{G}}\int_\Omega |Z(\mathbf{x})-c_1|^2 H(\phi_G(\mathbf{x})d\mathbf{x} + \\ \lambda_{2\mathrm{G}}\int_\Omega |Z(\mathbf{x})-c_2|^2 (1-H(\phi_G(\mathbf{x}))d\mathbf{x} + \\ \lambda_1\int_\Omega |Z(\mathbf{x})-c_1|^2 H(\phi_L(\mathbf{x})d\mathbf{x} + \\ \lambda_2\int_\Omega |Z(\mathbf{x})-c_1|^2 (1-H(\phi_L(\mathbf{x}))H(\phi_G(\mathbf{x})d\mathbf{x} + \\ \lambda_3\int_\Omega |Z(\mathbf{x})-c_2|^2 (1-H(\phi_L(\mathbf{x}))(1-H(\phi_G(\mathbf{x}))d\mathbf{x}.\end{aligned} \tag{6.17}$$

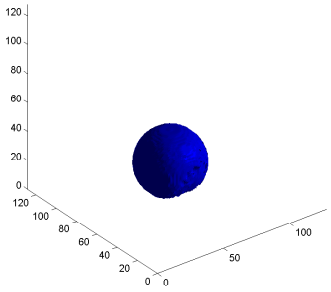
Here the parameters μ_1 , μ_2 , μ_L, μ_G, $\lambda_{1\mathrm{G}}$, $\lambda_{2\mathrm{G}}$, λ_1, λ_2, λ_3 are all positive. The two terms of the form $\int_\Omega(|\nabla\phi(\mathbf{x})|-1)^2 d\mathbf{x}$ are to automatically scale level set functions to avoid re-initialization for $\phi_L(\mathbf{x})$ and $\phi_G(\mathbf{x})$ following [92] and others.

Before we derive the solutions for (6.17), as usual, we replace the Heaviside function H by its regularized version H_ϵ. Moreover below for brevity, we use d, Z, ϕ_L, ϕ_G to denote $d(\mathbf{x})$, $Z(\mathbf{x})$, $\phi_L(\mathbf{x})$ and $\phi_G(\mathbf{x})$, respectively. To solve (6.17), keeping ϕ fixed and minimizing with respect to c_1 and c_2, we have the following equations for computing c_1 and c_2:

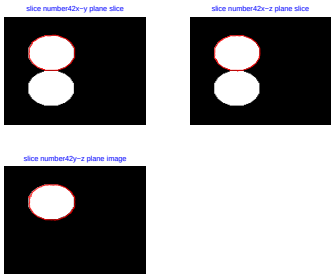
$$c_1 = \frac{\lambda_{1\mathrm{G}}\int_\Omega Z H_\epsilon(\phi_G)d\mathbf{x} + \lambda_1\int_\Omega Z H_\epsilon(\phi_L)d\mathbf{x} + \lambda_2\int_\Omega Z(1-H_\epsilon(\phi_L))H_\epsilon(\phi_G)d\mathbf{x}}{\lambda_{1\mathrm{G}}\int_\Omega H_\epsilon(\phi_G)d\mathbf{x} + \lambda_1\int_\Omega H_\epsilon(\phi_L)d\mathbf{x} + \lambda_2\int_\Omega(1-H_\epsilon(\phi_L))H_\epsilon(\phi_G)d\mathbf{x}}, \tag{6.18}$$



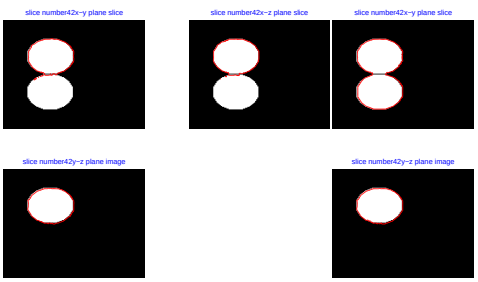
(a) Tested object



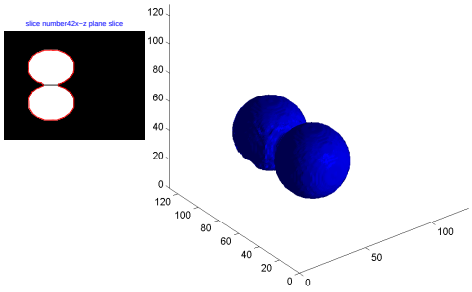
(b) Successful segmentation of one of the spheres with Badshah-Chen model [12] after 100 iterations



(c) Various slice of the segmented object with Badshah-Chen [12] model after 100 iteration



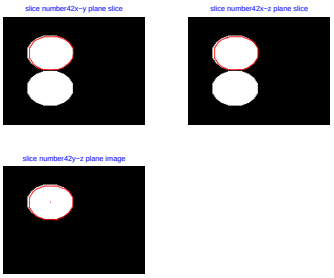
(d) Various slices of the segmented object with Badshah-Chen model [12] after 200 iteration



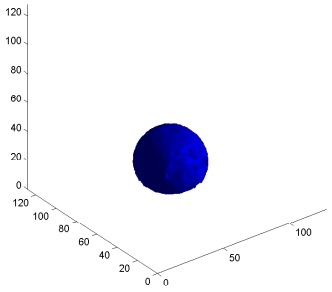
(e) Various slice of the segmented object with Badshah-Chen model [12] after 300 iteration

(f) Failed segmentation of one of the spheres with Badshah-Chen model [12]

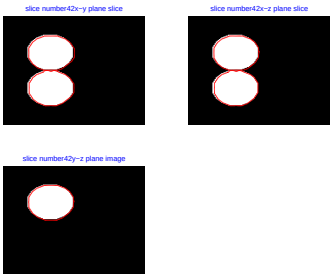
Figure 6.6: Failed segmentation by the Badshah-Chen model of a 3D object with two spheres. Here 4 markers are given within the surface of the aimed sphere and $\Delta t = 0.1$ for long term iterations with AOS method.



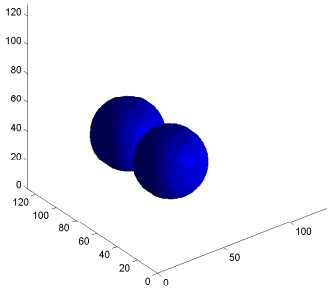
(a) Various slice of the segmented object with Gout-Guyader model [69] after 100 iterations



(b) selected object after 100 iteration with Gout-Guyader model [69]

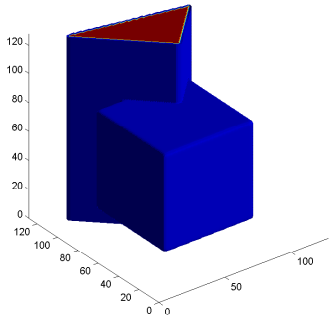


(c) Failed segmentation of one of the spheres with Gout-Guyader model [69] after 300 iterations

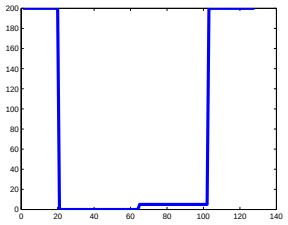


(d) Failed segmentation of one of the spheres with Gout-Guyader model [69] after 300 iterations

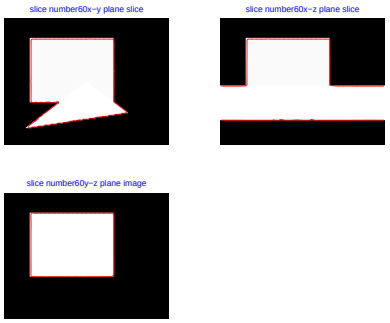
Figure 6.7: Failed segmentation by the Gout-Guyader model [69] of a 3-D object with two spheres in long term iterations. Here 4 markers are given within the surface of the aimed sphere and $\Delta t = 0.1$ for 300 AOS iterations.



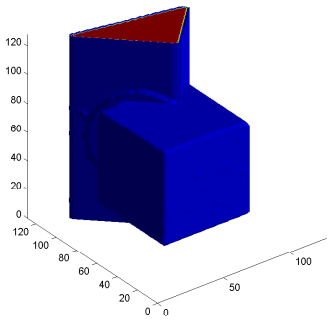
(a) Given 3-D image



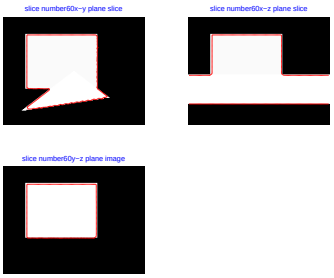
(b) plane cut view through the vertical in middle of the image



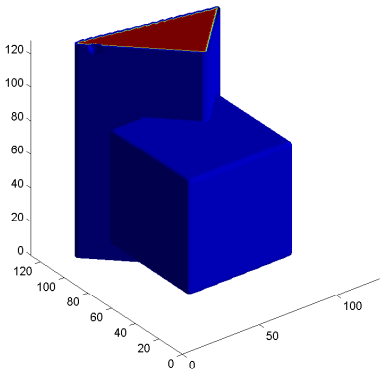
(c) Various slices of the segmented object with Badshah-Chen [12] from xy, yz, and xz direction



(d) Detected wrong object with Badshah-Chen model [12]



(e) Detected wrong object with Gout-Guyader model [69]



(f) Detected wrong object with Gout-Guyader model [69]

Figure 6.8: A challenging test image for the Badshah-Chen and Gout-Guyader models. Here the given image in (a) consists of a triangular prism over a cuboid and the two objects have a small intensity difference of 5 units. (b) shows a plane cut view through the vertical middle of the image. The model segments both the objects by failing to segment the aimed triangular prism with $\Delta t = 0.1$ and 50 AOS iterations.

$$c_2 = \frac{\lambda_{2G}\int_\Omega Z(1-H_\epsilon(\phi_G))d\mathbf{x} + \lambda_3\int_\Omega Z(1-H_\epsilon(\phi_L))(1-H_\epsilon(\phi_G))d\mathbf{x}}{\lambda_{2G}\int_\Omega (1-H_\epsilon(\phi_G))d\mathbf{x} + \lambda_3\int_\Omega (1-H_\epsilon(\phi_L))(1-H_\epsilon(\phi_G))d\mathbf{x}} \tag{6.19}$$

if we assume that the $\phi_G(\mathbf{x})$ has neither an empty interior nor an empty exterior. Now keeping c_1 and c_2 fixed, and denoting $W = dg(|\nabla Z|)$, we derive the Euler-Lagrange equations for solving $\phi_L(\mathbf{x})$ and $\phi_G(\mathbf{x})$ from (6.17) using the Gâteaux derivatives, as detailed in Section §5.3:

$$\begin{cases} \mu_1\delta_\epsilon(\phi_L)\nabla\cdot\left(WH_\epsilon(\phi_G)\dfrac{\nabla\phi_L}{|\nabla\phi_L|}\right) + \mu_L\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_L|})\nabla\phi_L\right)+ \\ \delta_\epsilon(\phi_L)\Big(-\lambda_1(Z(\mathbf{x})-c_1)^2 + \lambda_2(Z(\mathbf{x})-c_1)^2H_\varepsilon(\phi_G)+ \\ \quad \lambda_3(Z(\mathbf{x})-c_2)^2(1-H_\varepsilon(\phi_G))\Big) = 0, \quad \text{in } \Omega \\ \frac{\partial\phi_L}{\partial\vec{n}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{6.20}$$

where boundary conditions $\mu_1 WH_\epsilon(\phi_G)\frac{\delta_\epsilon(\phi_L)}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}} = 0$ and $\mu_{\mathrm{L}}(|\phi_L|-1)\frac{1}{|\nabla\phi_L|}\frac{\partial\phi_L}{\partial\vec{n}} = 0$ reduce to Neumann boundary condition.

In the same way we derive the following Euler-Lagrange equation for ϕ_G:

$$\begin{cases} \mu_2\delta_\epsilon(\phi_G)\nabla\cdot\left(g\dfrac{\nabla\phi_G}{|\nabla\phi_G|}\right) + \mu_G\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_G|})\nabla\phi_G\right)- \\ \delta_\epsilon(\phi_G)\mu_1 W\,|\nabla H_\varepsilon(\phi_L)|+ \\ \delta_\epsilon(\phi_G)\Big(-\lambda_{1G}(Z(\mathbf{x})-c_1)^2 + \lambda_{2G}(Z(\mathbf{x})-c_2)^2- \\ \lambda_2(Z(\mathbf{x})-c_1)^2(1-H(\phi_L)) + \lambda_3(Z(\mathbf{x})-c_2)^2(1-H(\phi_L))\Big) = 0, \quad \text{in } \Omega \\ \dfrac{\partial\phi_G}{\partial\vec{n}} = 0 \text{ on } \partial\Omega. \end{cases} \tag{6.21}$$

In both equations, balloon terms such as $\alpha W|\nabla\phi_L|$ and $\alpha g(\mathbf{x})|\nabla\phi_G|$ can be added to speed up the convergence. The final equations (6.20) and (6.21) will be rewritten as

$$\begin{cases} \mu_1\delta_\epsilon(\phi_L)\nabla\cdot\left(WH_\epsilon(\phi_G)\dfrac{\nabla\phi_L}{|\nabla\phi_L|}\right) + \mu_L\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_L|})\nabla\phi_L\right)+ \\ \delta_\epsilon(\phi_L)\Big(-\lambda_1(Z(\mathbf{x})-c_1)^2 + \lambda_2(Z(\mathbf{x})-c_1)^2H_\varepsilon(\phi_G)+ \\ \qquad \lambda_3(Z(\mathbf{x})-c_2)^2(1-H_\varepsilon(\phi_G))\Big) \;+\; \alpha W(\mathbf{x})|\nabla\phi_L| = 0, \quad \text{in } \Omega \\ \frac{\partial\phi_L}{\partial\vec{n}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{6.22}$$

and

$$\begin{cases} \mu_2\delta_\epsilon(\phi_G)\nabla\cdot\left(g(\mathbf{x})\dfrac{\nabla\phi_G}{|\nabla\phi_G|}\right)+\mu_G\nabla\cdot\left((1-\dfrac{1}{|\nabla\phi_G|})\nabla\phi_G\right)+ \\ \delta_\epsilon(\phi_G)\Big(-\mu_1 W(\mathbf{x})\,|\nabla H_\varepsilon(\phi_L)|\Big)+ \\ \delta_\epsilon(\phi_G)\Big(-\lambda_{1\mathrm{G}}(Z(\mathbf{x})-c_1)^2+\lambda_{2\mathrm{G}}(Z(\mathbf{x})-c_2)^2-\lambda_2(Z(\mathbf{x})-c_1)^2(1-H(\phi_L)) \\ \qquad +\lambda_3(Z(\mathbf{x})-c_2)^2(1-H(\phi_L)\Big)\ +\ \alpha g(\mathbf{x})|\nabla\phi_G|=0, \quad \text{in } \Omega \\ \frac{\partial\phi_G}{\partial\vec{n}}=0 \ \text{ on } \partial\Omega. \end{cases} \tag{6.23}$$

To get linearised systems of equations we freeze the nonlinear coefficients in (6.22) and (6.23) and then we can solve them by a fixed point method. As the drawback of this method is the computational cost of the associated linear system for large images, we develop an AOS algorithm, which is detailed in the subsection that follows.

6.4.2 The 3-D Dual Level Set AOS Algorithm

In order to develop an AOS algorithm [69, 171, 106, 175] for (6.22) and (6.23), we consider the following related parabolic equations:

$$\begin{cases} \dfrac{\partial\phi_L}{\partial t}=\mu_1\delta_\epsilon(\phi_L)\nabla\cdot(F_L\nabla\phi_L)+\mu_{\mathrm{L}}\nabla\cdot(E_L\nabla\phi_L)+f_L\ = \\ \qquad \mu_1\delta_\epsilon(\phi_L)(\partial_x(F_L\partial_x\phi_L)+\partial_y(F_L\partial_y\phi_L)+\partial_z(F_L\partial_z\phi_L))+ \\ \qquad \mu_{\mathrm{L}}(\partial_x(E_L\partial_x\phi_L)+\partial_y(E_L\partial_y\phi_L)+\partial_z(E_L\partial_z\phi_L))+f_L, \\ \dfrac{\partial\phi_G}{\partial t}=\mu_2\delta_\epsilon(\phi_G)\nabla\cdot(F_G\nabla\phi_G)+\mu_{\mathrm{G}}\nabla\cdot(E_G\nabla\phi_G)+f_G\ = \\ \qquad \mu_2\delta_\epsilon(\phi_G)(\partial_x(F_G\partial_x\phi_G)+\partial_y(F_G\partial_y\phi_G)+\partial_z(F_G\partial_z\phi_G))+ \\ \qquad \mu_{\mathrm{G}}(\partial_x(E_G\partial_x\phi_G)+\partial_y(E_G\partial_y\phi_G)+\partial_z(E_G\partial_z\phi_G))+f_G. \end{cases} \tag{6.24}$$

where, to derive a compact and self-adjoint form, we have denoted

$$\begin{aligned} f_L &= \delta_\epsilon(\phi_L)\Big(-\lambda_1(Z(\mathbf{x})-c_1)^2+\lambda_2(Z(\mathbf{x})-c_1)^2H_\varepsilon(\phi_G)\ + \\ &\qquad \lambda_3(Z(\mathbf{x})-c_2)^2(1-H_\varepsilon(\phi_G))\Big)+\alpha W(\mathbf{x})|\nabla\phi_L|, \\ f_G &= \delta_\epsilon(\phi_G)\Big(-\mu_1 W(\mathbf{x})H_\varepsilon(\phi_L)\Big)+ \\ &\qquad \delta_\epsilon(\phi_G)\Big(-\lambda_{1\mathrm{G}}(Z(\mathbf{x})-c_1)^2+\lambda_{2\mathrm{G}}(Z(\mathbf{x})-c_2)^2+\alpha g(\mathbf{x})|\nabla\phi_G|\ - \\ &\qquad \lambda_2(Z(\mathbf{x})-c_1)^2(1-H(\phi_L))+\lambda_3(Z(\mathbf{x})-c_2)^2(1-H(\phi_L))\Big), \\ F_L &= \frac{WH_\epsilon(\phi_G)}{|\nabla\phi_L|},\ F_G=\frac{g}{|\nabla\phi_G|}, \quad E_L=1-\frac{1}{|\nabla\phi_L|},\ E_G=1-\frac{1}{|\nabla\phi_G|}. \end{aligned}$$

Since ϕ_G and ϕ_L depend on each other and the coefficients contain the nonlinearities we have to iterate the above linearized equations. It suffices to consider how to solve

the second equation:

$$\frac{\partial \phi_G}{\partial t} = \mu_2 \delta_\epsilon(\phi_G)(\partial_x(F_G \partial_x \phi_G) + \partial_y(F_G \partial_y \phi_G) + \partial_z(F_G \partial_z \phi_G)) + \\ \mu_G(\partial_x(E_G \partial_x \phi_G) + \partial_y(E_G \partial_y \phi_G) + \partial_z(E_G \partial_z \phi_G)) + f_G. \tag{6.25}$$

Below we shall write ϕ, F, E after dropping the subscripts in ϕ_G, F_G, E_G. Then the equation can be rewritten in the semi-implicit form

$$3\phi^{n+1} = 3\phi^n + 3\Delta t\Big(A_1(\phi^n)\phi^{n+1} + A_2(\phi^n)\phi^{n+1} + A_3(\phi^n)\phi^{n+1}\Big) + 3\Delta t f(\mathbf{x})$$

where both sides have been multiplied with a factor of 3. Splitting in x, y and z directions, we obtain

$$\begin{cases} \phi^{n+1} = \phi^n + 3\Delta t A_1(\phi^n)\phi^{n+1} + \Delta t f(\mathbf{x}), \\ \phi^{n+1} = \phi^n + 3\Delta t A_2(\phi^n)\phi^{n+1} + \Delta t f(\mathbf{x}), \\ \phi^{n+1} = \phi^n + 3\Delta t A_3(\phi^n)\phi^{n+1} + \Delta t f(\mathbf{x}). \end{cases}$$

In matrix notation, the above can be written as

$$(\ I - 3\Delta t A_l(\phi^n))\phi_l^{n+1} = \phi^n + \Delta t f(\mathbf{x}) \text{ for } l = 1, 2, 3$$

which define the AOS solution

$$\phi^{n+1} = \frac{1}{3}\sum_{l=1}^{3} \phi_l^{n+1}.$$

Here A_l for $l = 1, 2, 3$ are tridiagonal matrices derived from central differences (CD) as follows:

$$(A_1(\phi^n)\phi^{n+1})_{i,j,k} = \mu\delta_\epsilon(\phi^n)\left(\partial_x(\frac{\widetilde{g}^n}{|\nabla\phi^n|}\partial_x\phi^{n+1})\right)_{i,j,k} + \mu_{\mathrm{GL}}\left(\partial_x(\frac{|\nabla\phi^n|-1}{|\nabla\phi^n|}\partial_x\phi^{n+1})\right)_{i,j,k} \overset{CD}{=} \\ \frac{F^n_{i+1/2,j,k}(\partial_x\phi^{n+1})_{i+1/2,j,k} - F^n_{i-1/2,j,k}(\partial_x\phi^{n+1})_{i-1/2,j,k}}{h_x} + \\ \frac{E^n_{i+1/2,j,k}(\partial_x\phi^{n+1})_{i+1/2,j,k} - E^n_{i-1/2,j,k}(\partial_x\phi^{n+1})_{i-1/2,j,k}}{h_x} \overset{CD}{=} \\ \frac{(\frac{\widetilde{g}^n}{|\nabla\phi^n|})_{i+1/2,j,k}(\frac{\phi^{n+1}_{i+1,j,k}-\phi^{n+1}_{i,j,k}}{h_x}) - (\frac{\widetilde{g}^n}{|\nabla\phi^n|})_{i-1/2,j,k}(\frac{\phi^{n+1}_{i,j,k}-\phi^{n+1}_{i-1,j,k}}{h_x})}{h_x} + \\ \frac{(\frac{\nabla\phi^n|-1}{|\nabla\phi^n|})_{i+1/2,j,k}(\frac{\phi^{n+1}_{i+1,j,k}-\phi^{n+1}_{i,j,k}}{h_x}) - (\frac{\nabla\phi^n|-1}{|\nabla\phi^n|})_{i-1/2,j,k}(\frac{\phi^{n+1}_{i,j,k}-\phi^{n+1}_{i-1,j,k}}{h_x})}{h_x} = \\ A_{1,i,j,k}(\phi^{n+1}_{i+1,j,k} - \phi^{n+1}_{i,j,k}) - A_{2,ij,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i-1,j,k}) + A^1_{1,i,j,k}(\phi^{n+1}_{i+1,j,k} - \phi^{n+1}_{i,j,k}) - \\ A^1_{2,ij,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i-1,j,k}),$$

$$(A_2(\phi^n)\phi^{n+1})_{i,j,k} = \left(\partial_y(\frac{\widetilde{g}^n}{|\nabla\phi^n|}\partial_y\phi^{n+1})\right)_{i,j,k} + \mu_{\mathrm{GL}}\left(\partial_y(\frac{|\nabla\phi^n|-1}{|\nabla\phi^n|}\partial_y\phi^{n+1})\right)_{i,j,k} \overset{CD}{=}$$
$$\frac{F^n_{i,j+1/2,k}(\partial_y\phi^{n+1})_{i,j+1/2,k} - F^n_{i,j-1/2,k}(\partial_y\phi^{n+1})_{i,j-1/2,k}}{h_y} +$$
$$\frac{E^n_{i,j+1/2,k}(\partial_y\phi^{n+1})_{i,j+1/2,k} - E^n_{i,j-1/2,k}(\partial_y\phi^{n+1})_{i,j-1/2,k}}{h_y} \overset{CD}{=}$$
$$\frac{(\frac{\widetilde{g}^n}{|\nabla\phi^n|})_{i,j+1/2,k}(\frac{\phi^{n+1}_{i,j+1,k}-\phi^{n+1}_{i,j,k}}{h_y}) - (\frac{\widetilde{g}^n}{|\nabla\phi^n|})_{i,j-1/2,k}(\frac{\phi^{n+1}_{i,j,k}-\phi^{n+1}_{i,j-1,k}}{h_y})}{h_y} +$$
$$\frac{(\frac{\nabla\phi^n|-1}{|\nabla\phi^n|})_{i,j+1/2,k}(\frac{\phi^{n+1}_{i,j+1,k}-\phi^{n+1}_{i,j,k}}{h_y}) - (\frac{\nabla\phi^n|-1}{|\nabla\phi^n|})_{i,j-1/2,k}(\frac{\phi^{n+1}_{i,j,k}-\phi^{n+1}_{i,j-1,k}}{h_y})}{h_y} =$$
$$B_{1,i,j,k}(\phi^{n+1}_{i,j+1,k} - \phi^{n+1}_{i,j,k}) - B_{2,i,j,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i,j-1,k}) + B^1_{1,i,j,k}(\phi^{n+1}_{i,j+1,k} - \phi^{n+1}_{i,j,k}) -$$
$$B^1_{2,i,j,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i,j-1,k}),$$

$$(A_3(\phi^n)\phi^{n+1})_{i,j,k} = \left(\partial_z(\frac{\widetilde{g}^n}{|\nabla\phi^n|}\partial_z\phi^{n+1})\right)_{i,j,k} + +\mu_{\mathrm{GL}}\left(\partial_z(\frac{|\nabla\phi^n|-1}{|\nabla\phi^n|}\partial_z\phi^{n+1})\right)_{i,j,k} \overset{CD}{=}$$
$$\frac{F^k_{i,j,k+1/2}(\partial_z\phi^{n+1})_{i,j,k+1/2} - F^k_{i,j-1/2}(\partial_z\phi^{n+1})_{i,j-1/2,k}}{h_z} +$$
$$\frac{E^k_{i,j,k+1/2}(\partial_z\phi^{n+1})_{i,j,k+1/2} - E^k_{i,j-1/2}(\partial_z\phi^{n+1})_{i,j-1/2,k}}{h_z} \overset{CD}{=}$$
$$\frac{(\frac{\widetilde{g}^n}{|\nabla\phi^n|})_{i,j,k+1/2}(\frac{\phi^{n+1}_{i,j,k+1}-\phi^{n+1}_{i,j,k}}{h_z}) - (\frac{\widetilde{g}^n}{|\nabla\phi^n|})_{i,j,k-1/2}(\frac{\phi^{n+1}_{i,jk}-\phi^{n+1}_{i,j,k-1}}{h_z})}{h_z} +$$
$$\frac{(\frac{|\nabla\phi^n|-1}{|\nabla\phi^n|})_{i,j,k+1/2}(\frac{\phi^{n+1}_{i,j,k+1}-\phi^{n+1}_{i,j,k}}{h_z}) - (\frac{|\nabla\phi^n|-1}{|\nabla\phi^n|})_{i,j,k-1/2}(\frac{\phi^{n+1}_{i,jk}-\phi^{n+1}_{i,j,k-1}}{h_z})}{h_z} =$$
$$C_{1,i,j,k}(\phi^{n+1}_{i,j,k+1} - \phi^{n+1}_{i,j,k}) - C_{2,i,j,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i,j,k-1}) + C^1_{1,i,j,k}(\phi^{n+1}_{i,j,k+1} - \phi^{n+1}_{i,j,k}) -$$
$$C^1_{2,i,j,k}(\phi^{n+1}_{i,j,k} - \phi^{n+1}_{i,j,k-1})$$

with

$$A_{1,i,j,k} = \frac{\widetilde{g}^n_{i,j,k} + \widetilde{g}^n_{i+1,j,k}}{2h_x^2} \frac{1}{\sqrt{(\phi_x^2)_{i+1/2,j,k} + (\phi_y^2)_{i+1/2,j,k} + (\phi_z^2)_{i+1/2,j,k} + \beta}},$$

$$A_{2,i,j,k} = \frac{\widetilde{g}^n_{i-1,j,k} + \widetilde{g}^n_{i,j,k}}{2h_x^2} \frac{1}{\sqrt{(\phi_x^2)_{i-1/2,jk} + (\phi_y^2)_{i-1/2,jk} + (\phi_z^2)_{i-1/2,jk} + \beta}},$$

$$B_{1,i,j,k} = \frac{\widetilde{g}^n_{i,j,k} + \widetilde{g}^n_{i,j+1,k}}{2h_y^2} \frac{1}{\sqrt{(\phi_x^2)_{i,j+1/2,k} + (\phi_y^2)_{i,j+1/2,k} + (\phi_z^2)_{i,j+1/2,k} + \beta}},$$

$$B_{2,i,j,k} = \frac{\widetilde{g}^n_{i,j-1,k} + \widetilde{g}^n_{i,j,k}}{2h_y^2} \frac{1}{\sqrt{(\phi_x^2)_{i,j-1/2,k} + (\phi_y^2)_{i,j-1/2,k} + (\phi_z^2)_{i,j-1/2,k} + \beta}},$$

$$C_{1,i,j,k} = \frac{\widetilde{g}^n_{i,j,k} + \widetilde{g}^n_{i,j,k+1}}{2h_z^2} \frac{1}{\sqrt{(\phi_x^2)_{ij,k+1/2} + (\phi_y^2)_{ij,k+1/2} + (\phi_z^2)_{ij,k+1/2} + \beta}},$$

$$C_{2,i,j,k} = \frac{\widetilde{g}^n_{i,j,k-1} + \widetilde{g}^n_{i,j,k}}{2h_z^2} \frac{1}{\sqrt{(\phi_x^2)_{ij,k-1/2} + (\phi_y^2)_{ij,k-1/2} + (\phi_z^2)_{ij,k-1/2} + \beta}}$$

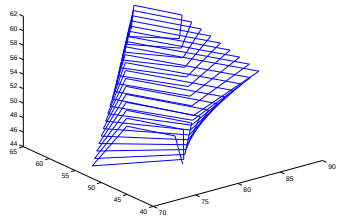
where $\widetilde{g}^n_{i,j,k} = g_{i,j,k}$ in the global case with ϕ_G and $\widetilde{g}^n_{i,j,k} = W_{i,j,k} H^n_{i,j,k}$ in the local case with ϕ_L. In the same way we calculate $A^1_{1,i,j,k}$, $A^1_{2,i,j,k}$, $B^1_{1,i,j,k}$, $B^1_{2,i,j,k}$, $C^1_{1,i,j,k}$ and $C^1_{2,i,j,k}$.

6.5 Experimental Results

In this section, we show 3 sets of experiments to comparison of the 3-D dual level set model with Badshah-Chen and Gout-Guyader models and demonstrate its robustness. The first set of 4 examples shows the success of the dual level set model for selecting an object among others for synthetic images. This is the same set of examples already processed with Badshah-Chen generalized to 3-D from the previous Subsection §6.3.3. The second set of 5 images shows the successfully segmented images with the new model for which the Badshah-Chen model and Gout-Guyader are unable to cope. The final set of 2 examples presents the results of using other segmentation models with input from our selective results.

In all experiments, we shall use the same parameters. These are $\lambda_1 = \lambda_2 = \lambda_{1cv} = \lambda_{2cv} = \lambda_{3cv} = 1$, $\tau = 4$, $\mu_1 = \mu_2 = n^2/10$, $\mu_L = \mu_G = 0.04$, $\alpha = 0.001$. For each case, we test two sizes $n = 128$ ($128 \times 128 \times 128$) and $n = 256$ ($256 \times 256 \times 256$), although we only present one of them.

The initial surface-contour is generally a sphere given by markers set such that the centre coordinate and radius are $\mathbf{w_0} = (x_0, y_0, z_0) = (\frac{1}{n_1}\sum_{i=1}^{n_1} x_i^*, \frac{1}{n_1}\sum_{i=1}^{n_1} y_i^*, \frac{1}{n_1}\sum_{i=1}^{n_1} z_i^*)$ and $r = \min_i |\mathbf{w}_i - \mathbf{w}_0|$ respectively, where $\mathcal{A} = \{\mathbf{w}_i^* = (x_i^*, y_i^*, z_i^*) \in \Omega, \quad 1 \le i \le n_1\} \subset \Omega$ is marker set. In challenging cases like a CT image with complex and non-convex shapes, we recommend to start with an initial contour-surface generated by a polytetrahedron. In this case we define the markers in three different layers only and construct a poly-tetrahedra (based on the given markers). See Figure 6.9 for one example.



(a)

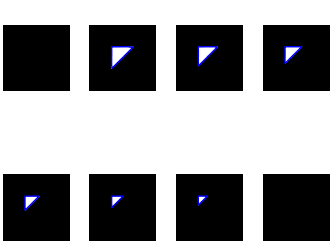
Figure 6.9: Contour-surface generated by a polytetrahedron.

6.5.1 Test Set 1 — robustness of the 3-D dual level set selective method

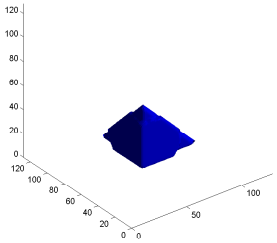
The first set of tests respectively in Figures 6.10, 6.11, 6.12 and 6.13 all consist of artificial geometrical objects, similarly to the Badshah-Chen model, we can see that the new dual level set model gives the same results. However we can notice that the new model is less sensitive to the choice of parameters, especially for ϵ.

6.5.2 Test Set 2 — comparison of 3-D segmentation of harder problems

In this set of 5 test problems we consider harder and more challenging 3-D cases. In the above set of examples we observe that the target feature/object is well separated from the other nearby ones. A challenging test would be when the distance of the objects or the intensity difference from others is small (see failed Badshah-Chen and Gout-Guyader models, Figure 6.8). The first example shown in Figure 6.14 presents a correctly segmented result by the 3-D dual level set model for the object presented in Figure 6.8, that failed with the Badshah-Chen model. In medical imaging such as CT and MRI, often, the whole image has low contrast so this small intensity difference poses a great challenge to selective segmentation models. The Badshah-Chen and Gout-Guyader models give reasonable segmentation at early iterations, such as Figure 6.15, and totally fail in other cases, such as Figure 6.14, 6.16 and 6.17. All of these 4 examples are correctly segmented in Figure 6.14, 6.15, 6.16, and 6.17 respectively, using the 3-D dual level set selective segmentation. One more example showing the practical importance of accurate selective segmentation in 3-D measuring of medical data is shown in Figure 6.18 where the cancer detection in the brain has been successfully segmented.

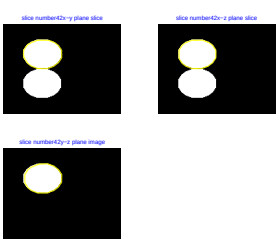


(a) Various slices of the segmented object from xy, yz, and xz direction

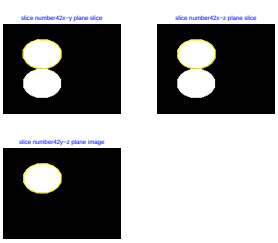


(b) Selected object with the new method

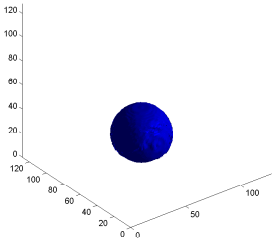
Figure 6.10: Set 1 Example 1: Successful detection of the pyramid in a clean and synthetic 3D image with 4 markers with 3-D dual level set method. Initial level set is a sphere within the center of the markers and $n = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$ or $\epsilon = 10^{0}$ after 100 iterations.



(a) Various slices of the segmented object from xy, yz, and xz direction after 100 iterations

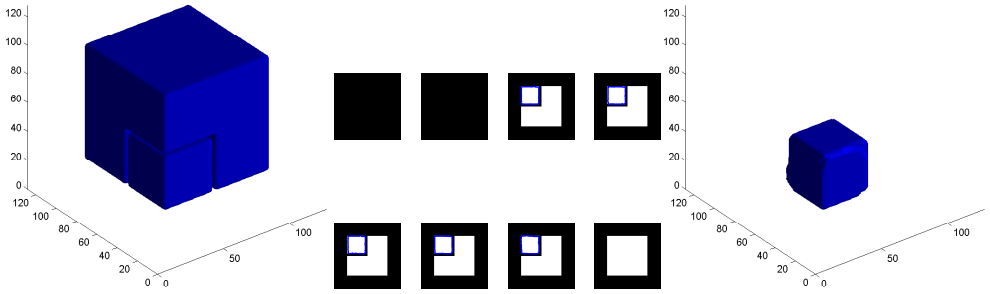


(b) Various slices of the segmented object from xy, yz, and xz direction after 300 iterations



(c) Selected object

Figure 6.11: Set 1 Example 2: Successful detection of one sphere out of two in a clean and synthetic 3-D image with 4 markers with 3-D dual level set method. Initial level set is a sphere within the centre of the markers and $n = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$ or $\epsilon = 10^{0}$ after 100 and 300 iterations.

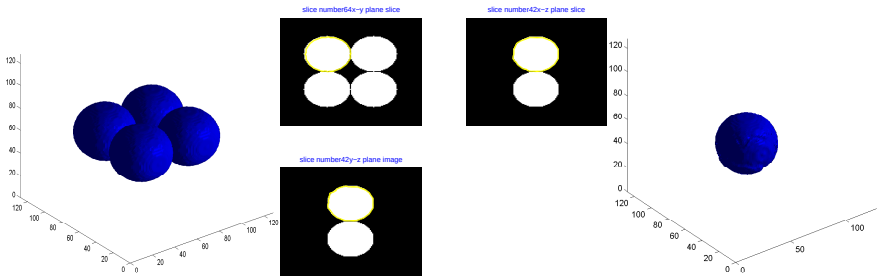


(a) Tested object

(b) Various slices of the segmented object from xy, yz, and xz direction

(c) Detected object with new method

Figure 6.12: Set 1 Example 3: Successful detection of the small cuboid in a clean and synthetic 3-D image with 4 markers with 3-D dual level set method. Initial level set is a sphere within the centre of the markers and $n = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$ or $\epsilon = 10^{0}$ after 100 iterations.

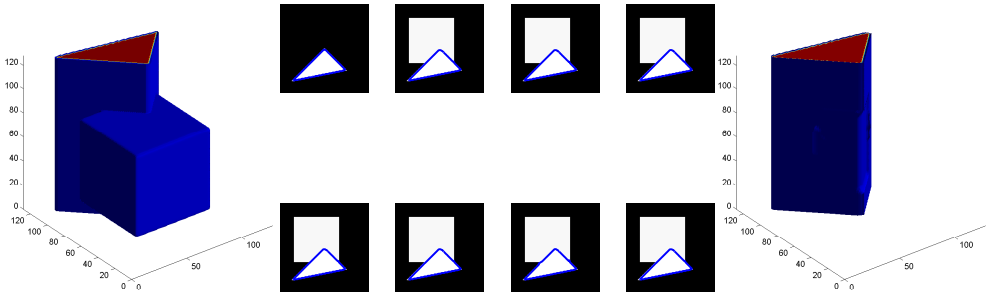


(a) Given 3-D volumetric with 4 spheres

(b) Various slices of the segmented object from xy, yz, and xz direction

(c) Selected segmented object

Figure 6.13: Set 1 Example 4: Successful detection of one sphere out of 4 in a clean and synthetic 3-D image with 4 markers with 3-D dual level set method. Initial level set is a sphere within the centre of the markers and $n = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$ or $\epsilon = 10^{0}$ after 100 (same results for more number of iterations).



(a) The given image (b) Various slices of the segmented object in xy direction with the new method (c) Detected object with the new method

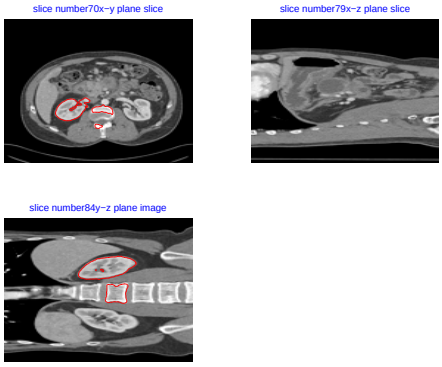
Figure 6.14: Set 2 Example 1: Successful detection of the prism in a clean and synthetic 3-D image with 6 markers with 3-D dual level set method while the Badshah-Chen [12] fails. Parameters are $n = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$ (or $\epsilon = 10^{0}$).

6.5.3 Test Set 3 — useful applications of a 3-D selection model

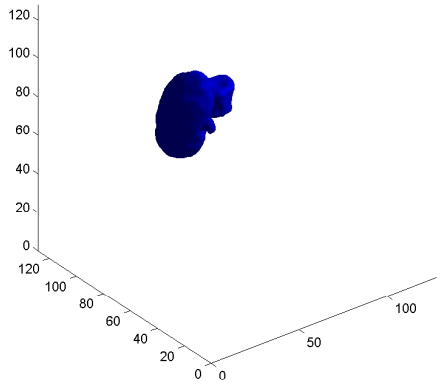
After a 3-D organ or object has been segmented with our selective segmentation method, we can look inside of this organ for a deeper segmentation. For example in the case of the kidney we can segment its inside structures and see what the veins looks like. We can do this by either the Chan-Vese model or Li-Kao-Gore-Ding model [91, 90]. The main difference is that the two piecewise constants c_1 and c_2 in Chan-Vese have been replaced by two piecewise smooth functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, fitting the image intensities near each point $\mathbf{x}$ in a local way. The experimental results show that both models succeed when the background intensity is equal to the mean of the object intensity, but if the background intensity is equal to 0, only Li-Kao-Gore-Ding model can succeed in both 2-D and 3-D cases, see Figure 6.19(d). In Figure 6.19 we show the results in a 2-D slice after the image has been processed with a selective segmentation model and the kidney has been captured. A full view of the 3-D case is shown in Figure 6.20 and Figure 6.21. Clearly our new selective model provides useful input to the deep segmentation models.

6.6 Conclusions

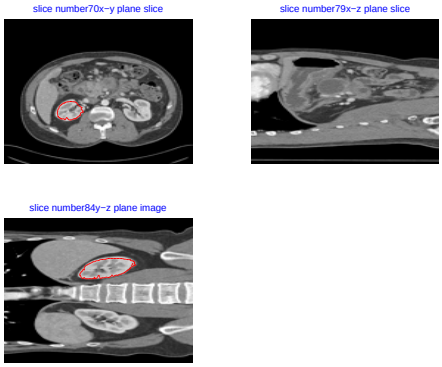
We have developed two new selective segmentation 3-D models for selective segmentation. The first model is an extension of the 2-D Badshah-Chen model based on a single level set function and the second new model is based on dual level set selective segmentation functions. The results indicate that the latter model is more robust or images with nearby objects having similar intensity values. The issues of refinements to the model to have more flexibility in geometric constraints and fast solution will be considered in Chapter 8.



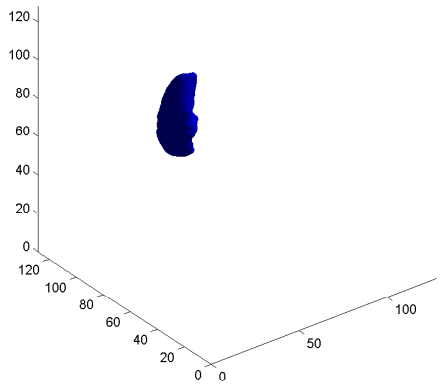
(a) Various slices in xy, yz and xz direction after segmentation with Badshah-Chen [12]



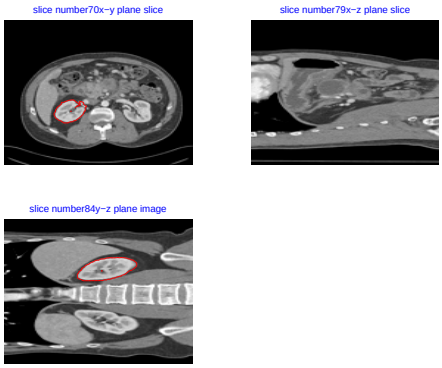
(b) Badshah-Chen model [12] accurate segmentation but failed due to spurious detections



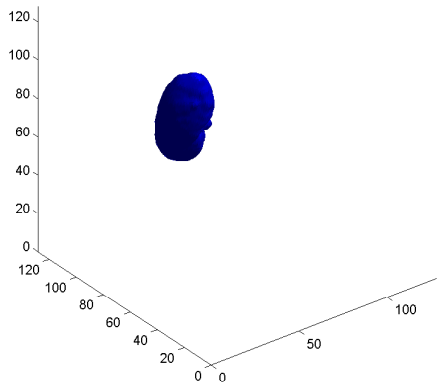
(c) Reasonable but inaccurate segmentation of Gout-Guyader model [69] due to noise



(d) Reasonable but inaccurate segmentation of Gout-Guyader model [69] due to noise

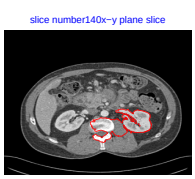


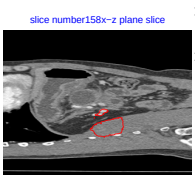
(e) Various slice in xy, yz and xz direction after segmentation with the new method

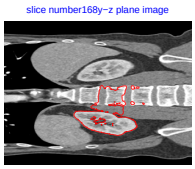


(f) Detected object with the new method

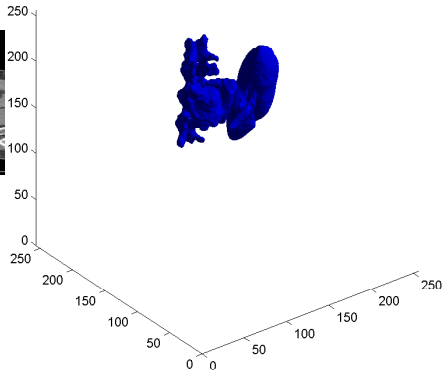
Figure 6.15: Set 2 Example 2: Successful detection of the left kidney in a 3-D CT volume data with 3-D dual level set method. Initial level set is a polyhedron constructed from 5 given markers in 3 different layers and $\eta = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$.



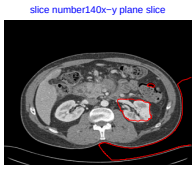


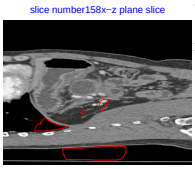


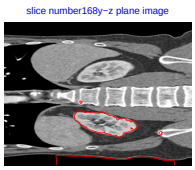
(a) Various slices in xy, yz and xz direction after segmentation with B-Ch [12]



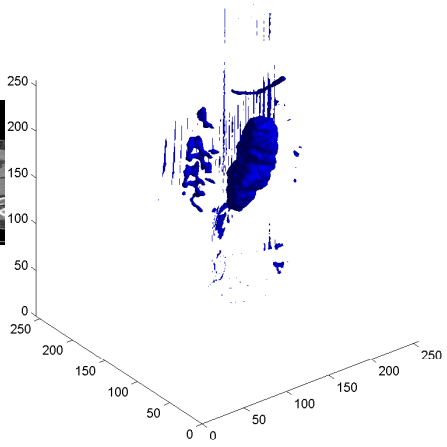
(b) B-Ch model [12] failed to detect the object



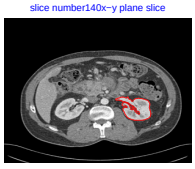


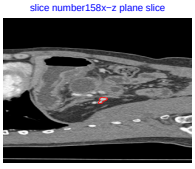


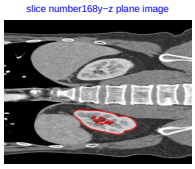
(c) Various slice in xy, yz and xz direction after segmentation with G-G [69]



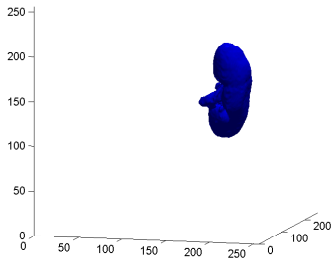
(d) G-G model [69] failed to detect the object





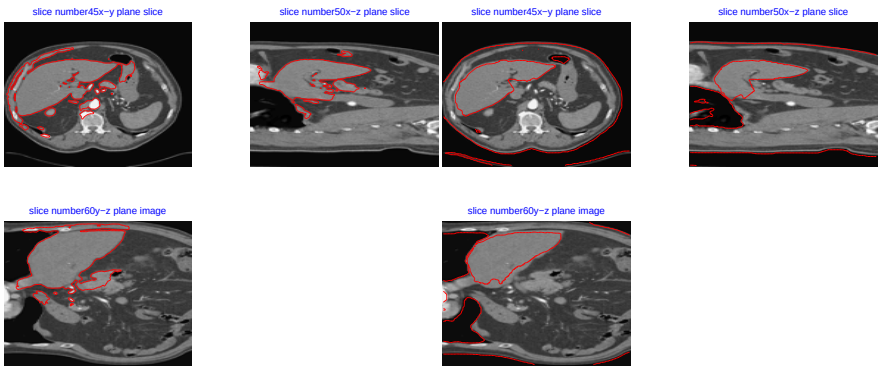


(e) Various slice in xy direction



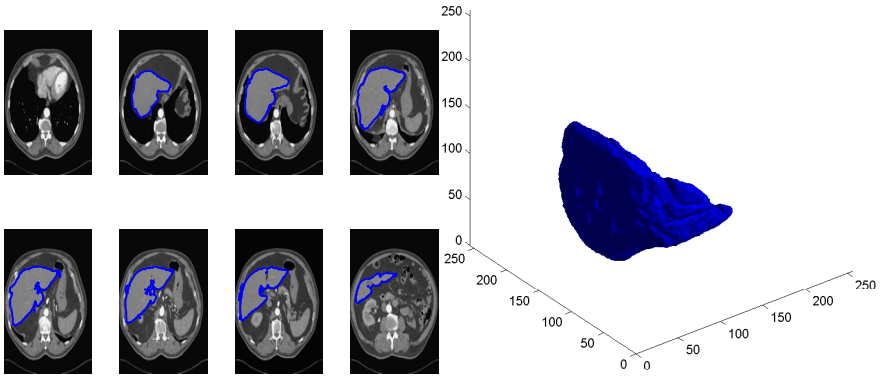
(f) Detected object

Figure 6.16: Set 2 Example 3: Successful detection of the right kidney in a 3-D CT volume data with 3-D dual level set method. Initial level set is a polyhedron constructed from 5 given markers in 3 different layers and $n = 128$, $dt = 0.1$, $\epsilon = 10^{-2}$ after 200 iterations.



(a) Failed segmentation with B-C model [12]

(b) Failed segmentation with G-G model [69]



(c) Slices in xy direction after segmentation with dual level set model

(d) Successful detected object with dual level set

Figure 6.17: Set 2 Example 4: Successful detection of liver in a 3-D CT volume data with 3-D dual level set method, while the old models fail. Initial level set is a polyhedron constructed from 5 given markers in 4 different layers and $n = 256$, $dt = 0.1$, $\epsilon = 10^{-2}$ after 200 iterations.

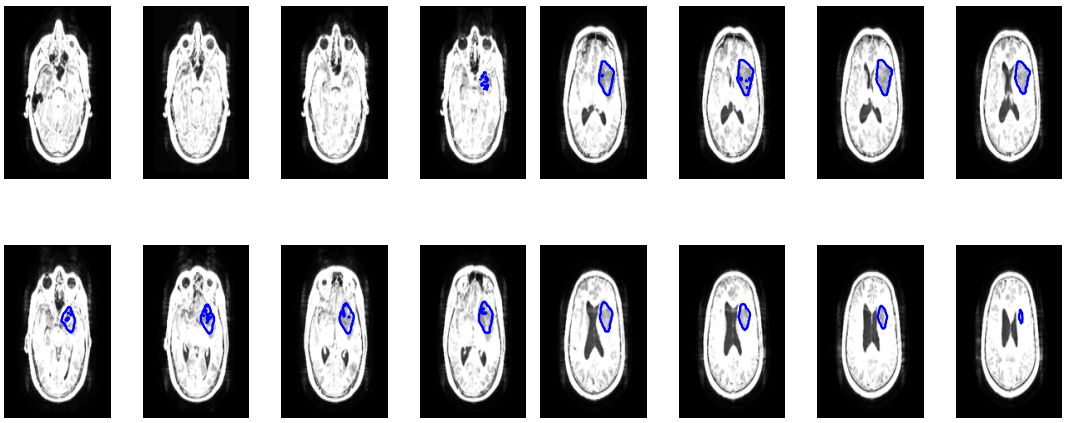
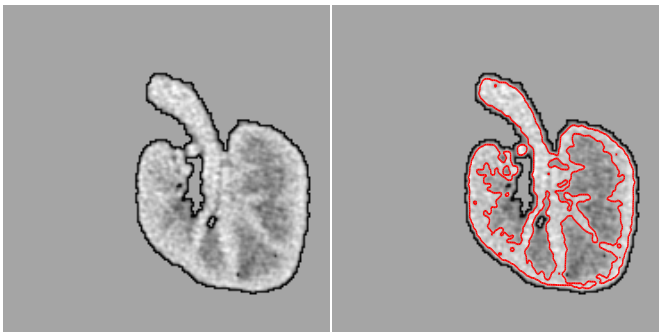
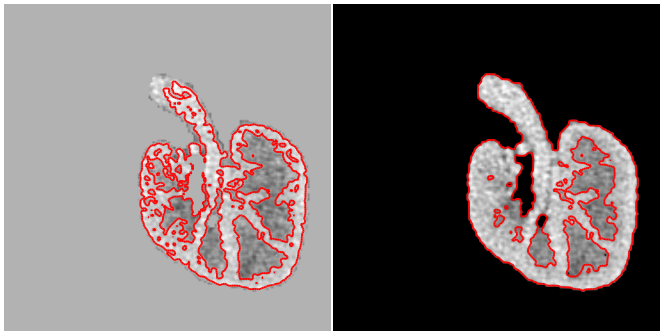


Figure 6.18: Set 2 Example 5: Slice in xy direction after segmentation of brain CT image with low glioma with 3-D dual level set. Successful detection of glioma with a polyhedron level set constructed with 4 given markers in each of the three different layers using the 3-D Dual Level Set method. The parameters are $n = 128$, $dt = 0.01$, $\epsilon = 10^{-2}$ after 100 iterations.



(a) Given selective image with background intensity equal to the mean of the object intensity

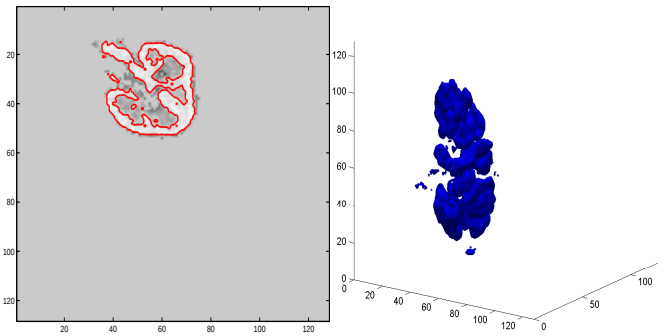
(b) Deep segmentation with Chan-Vese model



(c) Deep segmentation with Li-Kao-Gore-Ding model

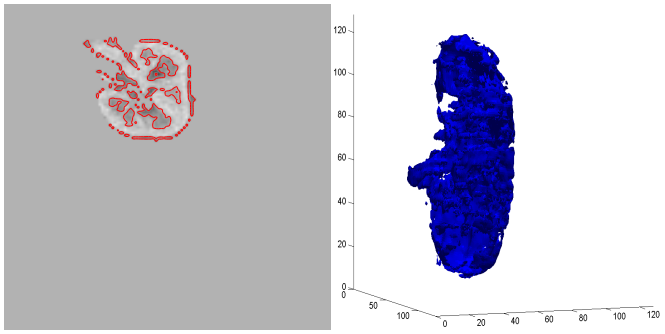
(d) Deep segmentation with Li-Kao-Gore-Ding model of the selected object with background intensity equal to 0

Figure 6.19: Set 3 Example 1: CT image post processed with Chan-Vese and Li-Kao-Gore-Ding model.



(a) 2-D slice of 3-D deep segmentation processed with C-V model

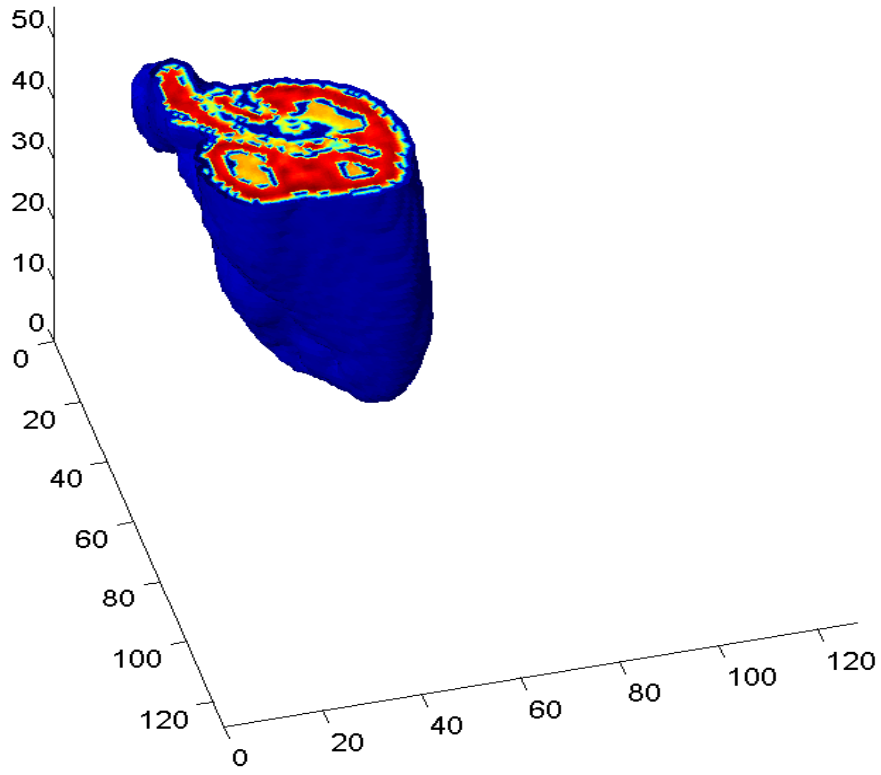
(b) 3-D shape of the deep segmentation with C-V model



(c) 2-D slice of 3-D deep segmentation processed with Li-Kao-Gore-Ding model

(d) 3-D shape of the deep segmentation with Li-Kao-Gore-Ding model

Figure 6.20: Set 3 Example 1: CT 3-D volume image post processed with Chan-Vese and Li-Kao-Gore-Ding model. Blood flow into veins has been distinguished.



(a) 3-D view of blood veins flow segmentation of the kidney

Figure 6.21: Set 3 Example 2: 3-D volume image post processed with Chan-Vese and Li-Kao-Gore-Ding models. In red we can see the segmentation obtained which shows the blood flow of the selected kidney.

Chapter 7

On a Variational Model for Selective Image Segmentation of Features with Infinite Perimeter

The variational models already detailed in the previous chapters are based on the minimization of the length of a feature's boundaries (i.e. $\mathcal{H}^1$ Hausdorff measure). However there exist problems with irregular and oscillatory object boundaries, where minimizing such a length is not appropriate, as noted by Barchiesi et al. [14] who proposed to minimize the $\mathcal{L}^2$ Lebesgue measure of the γ-neighbourhood of the boundaries instead. This chapter presents a dual level set selective segmentation model which improves the selective model in cases of real life images with oscillatory boundaries. We will show qualitative results demonstrating the effectiveness of the proposed method for these cases.

7.1 Introduction

In this chapter we will show a way of improving the 2-D selective segmentation work already introduced in Chapter 5. The model deals with the challenging task of selecting one feature/object. For the dual level set model it has been noticed that for proper incorporation of some geometric prior information (markers) into the image, edge detection techniques and a proper fitting term for the local level set could lead to good convergence to the target object. However, these models might not give good results in cases where the image has oscillatory boundaries since all techniques based on the weighted penalization term of the length can give rise to some difficulties for these cases.

In this chapter, we make use of the $\mathcal{L}^2$ Lebesgue measure of the γ-neighbourhood of the contour [14] as a penalization term instead of the $\mathcal{H}^1$ Hausdorff measure and propose a local selective segmentation model enabling us to extract features with irregular and oscillatory boundaries. The difference between these measures can be demonstrated with a simple example:

Given $\Omega = (0,1)^2$, $u_0(x,y)$ is defined as

$$u_0(x,y) = \begin{cases} 1 & y < 1 + \frac{x}{2} sin(\frac{1}{x}), \\ 0 & \text{otherwise.} \end{cases}$$

for all $(x, y) \in \Omega$. In this case, the contour is given by $\Gamma = (x, y) : y = 1 + \frac{x}{2} sin(\frac{1}{x})$ as shown in Fig. 7.1. It can be easily noticed that $\mathcal{L}(\gamma\text{-}\Gamma) \leq 1, \forall \gamma > 0$, while a simple calculation shows that $\mathcal{H}^1(\Gamma) = \infty$.

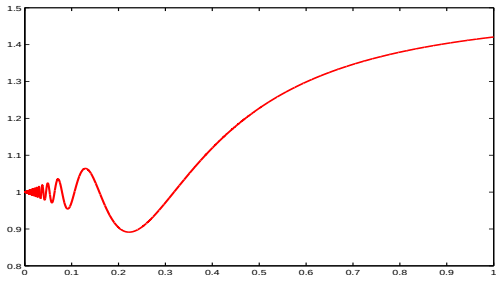


Figure 7.1: Illustration of an infinite perimeter function.

The remainder of this chapter is organized as follows. In Section §7.2, we describe the Barchiesi-Kang-Le-Morini-Ponsiglione model [14], which uses the $\mathcal{L}^2$ Lebesgue measure of the γ-neighbourhood of the contour as the regularization term of their variational model. A new dual level-set selective model, which uses the $\mathcal{L}^2$ Lebesgue measure of the γ-neighbourhood of the contour as a penalization term, is presented in Section §7.3. Time marching and AOS techniques are developed for the numerical solution. In Section §7.4 the experimental results are presented based on artificial and real images. Conclusions are drawn in Section §7.5.

7.2 A Variational Model for Infinite Perimeter Segmentation by Barchiesi-Kang-Le-Morini-Ponsiglione

Following the variational framework of Chan-Vese [37], see §3.4.7, while attempting to segment an object with irregular boundaries, Barchiesi et al.[14] proposed a variational segmentation model replacing the Chan-Vese length term $\mathcal{H}^1(\Gamma)$ with the area of the γ-neighbourhood of the edge set Γ

$$\gamma\text{-}\Gamma := \bigcup_{x \in \Gamma} B_\gamma(x), \tag{7.1}$$

where $B_\gamma(x)$ represents a γ-neighbourhood of the point x.

For a given bounded set $\Omega \subset R^2$ representing the image domain, the energy proposed is

$$\mathcal{F}_{L2}(\Gamma, c_1, c_2) = \mathcal{L}^2(\gamma\text{-}\Gamma) + \lambda \int_{\text{inside}(\Gamma)} |u_0(x,y) - c_1|^2 dxdy + \lambda \int_{\text{outside}(\Gamma)} |u_0(x,y) - c_2|^2 dxdy \tag{7.2}$$

where $\mathcal{L}^2$ is the 2-dimensional Lebesgue measure, and c_1 and c_2 are defined as with the Chan-Vese model.

The aim is to capture rough boundaries of the main objects in the image u_0 while at the same time achieving the denoising effect. In this way the $\mathcal{L}^2$ measure will help to preserve the finely oscillating boundaries of the main objects in the image and at the same time the model will allow for infinite perimeter segmentation.

Considering $f_0 := \chi_{[0,1]}$, and a smooth version of it f, a positive decreasing function such as $f(t) = e^{-t^k}$, with k a large and even number, or $f(t) = \frac{1}{1+t^k}$ for $k \geq 1$, the

$\mathcal{L}^2(\gamma\text{-}\Gamma)$ term can be rewritten as:

$$\mathcal{L}^2(\gamma\text{-}\Gamma) := \int_\Omega f_0(\frac{dist(x,\Gamma)}{\gamma})dx \approx \int_\Omega f(\frac{dist(x,\Gamma)}{\gamma})dx. \tag{7.3}$$

Similar to the previous chapters we will consider the level set formulation of this minimization problem. Considering the level set functions ϕ, a signed distance function from its zero level set Γ, we have

$$f(\frac{dist(\cdot,\Gamma)}{\gamma}) = f(\frac{|\phi|}{\gamma}).$$

Remark 7.2.1 *It is known that for a Γ regular curve (e.g., a smooth curve) the Minkowski content $\lim_{\gamma\longrightarrow 0}\frac{\mathcal{L}^2(\gamma-\Gamma)}{2\gamma}$ exists and coincides with the usual 1-dimensional measure $\mathcal{H}^1(\Gamma)$. This means that the first term of the energy function will be an approximation of the classical perimeter and on the other hand, for fixed γ, $\mathcal{L}^2(\gamma\text{-}\Gamma)$ is smaller than $2\gamma\mathcal{H}^1(\Gamma)$. For more details see [14] and the references therein.*

The model requires ϕ to look like a signed distance function. To do this we can add penalization terms, denoted by $P(\phi)$ which forces $|\nabla\phi| \leq 1$. As discussed in §2.6.3, there are different ways that we can force ϕ to look like a signed distance function. In the literature we often find the term $P(\phi) = \int_\Omega(|\nabla\phi| - 1)^2dx$ (see [92]), $\frac{1}{p}\int_\Omega|\nabla\phi|^p dx$ (for large p) or $\Delta_\infty\phi = \frac{1}{|\nabla\phi|^2}\sum_{i,j=1}^m \phi_{x_ix_j}\phi_{x_i}\phi_{x_j}$ (suggested in Barchiesi et al. [14, 118]), or a local image fitting energy functional, which can be viewed as a constraint of the differences between the fitting image and the original image (see [181]).

Thus by adding the signed distance function penalization term, the revised functional to be minimized in terms of the level set is the following

$$\mathcal{F}_{L2f}(\phi) = \int_\Omega f(\frac{|\phi|}{\gamma})+P(\phi)+\lambda_1\int_\Omega|u_0-c_1|^2H_\epsilon(\phi)dxdy+\lambda_2\int_\Omega|u_0-c_2|^2(1-H_\epsilon(\phi))dxdy. \tag{7.4}$$

Considering $f(t) = e^{-t^k}$, the above minimisation function (7.4) brings us to the following equation

$$\frac{k}{\gamma}\phi^{k-1}e^{-\frac{\phi}{\gamma}^k} + D_P(\phi) - \delta_\epsilon(\phi)\big[\lambda_1(u_0-c_1)^2 - \lambda_2(u_0-c_2)^2\big] = 0, \tag{7.5}$$

where $D_P(\phi)$ is the term derived from $P(\phi)$ (see §5.4), $\delta(\phi)$ is the approximation of the Dirac delta function, k is a large even number.

Solving equation (7.5) by a gradient descent method one would have

$$\frac{\partial\phi}{\partial t} = \frac{k}{\gamma}\phi^{k-1}e^{-\frac{\phi}{\gamma}^k} + D_P(\phi) - \delta_\epsilon(\phi)\big[\lambda_1(u_0-c_1)^2 - \lambda_2(u_0-c_2)^2\big]. \tag{7.6}$$

Through some mathematical analysis and tests, Barchiesi et al. [14] showed that their model has the capability of removing the noise, the cornering effect, resolution and capability of keeping oscillatory parts of the boundaries and also performs better in comparison with the Chan-Vese model [39] for these cases.

7.3 A New Dual Level Set Model for Infinite Perimeter

In this section we develop a selective segmentation dual level set method which obtains satisfactory segmentation results for image features with oscillatory boundaries. By equipping our variational model detailed in Chapter 5 with a new regularization term based on a γ-neighbourhood area of the contour Γ one expects to solve the problem of oscillatory boundaries and get more accurate results for these cases. For this reason, as a direct implication the replacement of the weighted length term in the dual level set model in equation (5.11) with the area of the γ-neighbourhood area of the edge set Γ is the key.

Recalling from Chapter 5 the dual level set selective segmentation model detects the features that are defined in a closed domain Ω and to be closest to the defined geometrical set of points $\mathcal{A}$, where $\mathcal{A} = \{w_i^* = (x_i^*, y_i^*) \in \Omega, \quad 1 \leq i \leq n_1\} \subset \Omega$, consisting of n_1 distinct points near the target object. The quantities $d(x, y)$ and $g(x, y)$ given in equations (5.2) and (5.1), respectively, are distance and edge detection functions with the property that they approach zero when near the Γ boundary and are large when away from it, see [12, 67, 69, 129, 179]. Those two functions are used as a stopping function for the curve evolution. In our new infinite perimeter model we consider $\mathcal{L}^2(\gamma\text{-}\Gamma) \approx \int_\Omega e^{-\frac{\phi^k}{\gamma^k}}$, which is an approximation of the γ-neighbourhood area of the contour in a given image $u_0(x, y)$.

7.3.1 The New Infinite Perimeter Dual Level Set Model

Let us assume that our image $u_0(x, y)$ has features with irregular and oscillatory boundaries. Replacing $\mathcal{H}^1$ (length term which is a Hausdorff measure) with $\mathcal{L}^2$ (Lebesgue measure) in equation (5.11) the new model's energy function can be written as follows in terms of the level set function

$$
\begin{aligned}
\min_{\phi_L(x,y),\phi_G(x,y),c_1,c_2} & \mathcal{F}_{IDLSS}(\phi_L, \phi_G, c_1, c_2) = \\
& \mu_1 \int_\Omega d(x,y) g(\nabla u_0) f_0(\frac{|\phi_L|}{\gamma}) H(\phi_G) + \frac{\mu_{\mathrm{L}}}{2} \int_\Omega (|\nabla \phi_L| - 1)^2 dxdy + \\
& \mu_2 \int_\Omega g(\nabla u_0) f_0(\frac{|\phi_G|}{\gamma}) + \frac{\mu_{\mathrm{G}}}{2} \int_\Omega (|\nabla \phi_G(x,y)| - 1)^2 dxdy + \\
& \lambda_{1\mathrm{G}} \int_\Omega |u_0(x,y) - c_1|^2 H(\phi_G(x,y) dxdy + \\
& \lambda_{2\mathrm{G}} \int_\Omega |u_0(x,y) - c_2|^2 (1 - H(\phi_G(x,y)) dxdy + \\
& \lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H(\phi_L(x,y) dxdy + \\
& \lambda_2 \int_\Omega |u_0(x,y) - c_1|^2 (1 - H(\phi_L(x,y)) H(\phi_G(x,y) dxdy + \\
& \lambda_3 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H(\phi_L(x,y)) (1 - H(\phi_G(x,y)) dxdy.
\end{aligned}
\tag{7.7}
$$

For numerical purposes, we replace the discontinuous function $f_0 := \chi_{[0,1]}$ with the smoother function $f(t) := e^{-t^k}$ which behaves more like f_0 for large k. This leads to

the following energy equation:

$$\begin{aligned}
&\min_{\phi_L(x,y),\phi_G(x,y),c_1,c_2} \mathcal{F}_{IDLSS}(\phi_L,\phi_G,c_1,c_2) = \\
&\mu_1 \int_\Omega d(x,y) g(\nabla u_0) e^{-(\frac{\phi_L}{\gamma})^k} H(\phi_G) + \frac{\mu_{\mathrm{L}}}{2} \int_\Omega (|\nabla \phi_L| - 1)^2 dxdy + \\
&\mu_2 \int_\Omega g(\nabla u_0) e^{-(\frac{\phi_G}{\gamma})^k} + \frac{\mu_{\mathrm{G}}}{2} \int_\Omega (|\nabla \phi_G(x,y)| - 1)^2 dxdy + \\
&\lambda_{1\mathrm{G}} \int_\Omega |u_0(x,y) - c_1|^2 H(\phi_G(x,y) dxdy + \\
&\lambda_{2\mathrm{G}} \int_\Omega |u_0(x,y) - c_2|^2 (1 - H(\phi_G(x,y)) dxdy + \\
&\lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H(\phi_L(x,y) dxdy + \\
&\lambda_2 \int_\Omega |u_0(x,y) - c_1|^2 (1 - H)(\phi_L(x,y)) H(\phi_G(x,y) dxdy + \\
&\lambda_3 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H)(\phi_L(x,y))(1 - H(\phi_G(x,y)) dxdy
\end{aligned} \tag{7.8}$$

where ϕ_L, ϕ_G, c_1, c_2, μ_1, μ_2, $\lambda_{1\mathrm{G}}$, $\lambda_{2\mathrm{G}}$, μ_{L}, μ_{G}, λ_1, λ_2, λ_3, $H(\phi_L)$, $H(\phi_G)$ are defined as for (5.11). Here the terms multiplied with $\lambda_1, \lambda_2, \lambda_3$ allow the total freedom of the level set functions so that ϕ_L will not pick up spurious features nearby.

Replacing the Heaviside function with a regularized Heaviside function we rewrite equation (7.8) as

$$\begin{aligned}
&\min_{\phi_L(x,y),\phi_G(x,y),c_1,c_2} \mathcal{F}_{IDLSS}(\phi_L,\phi_G,c_1,c_2) = \\
&\mu_1 \int_\Omega d(x,y) g(\nabla u_0) e^{-(\frac{\phi_L}{\gamma})^k} H_\epsilon(\phi_G) dxdy + \frac{\mu_{\mathrm{L}}}{2} \int_\Omega (|\nabla \phi_L| - 1)^2 dxdy + \\
&\mu_2 \int_\Omega g(\nabla u_0) e^{-(\frac{\phi_G}{\gamma})^k} dxdy + \frac{\mu_{\mathrm{G}}}{2} \int_\Omega (|\nabla \phi_G(x,y)| - 1)^2 dxdy + \\
&\lambda_{1\mathrm{G}} \int_\Omega |u_0(x,y) - c_1|^2 H_\epsilon(\phi_G(x,y) dxdy + \\
&\lambda_{2\mathrm{G}} \int_\Omega |u_0(x,y) - c_2|^2 (1 - H_\epsilon(\phi_G(x,y)) dxdy + \\
&\lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H_\epsilon(\phi_L(x,y) dxdy + \\
&\lambda_2 \int_\Omega |u_0(x,y) - c_1|^2 (1 - H_\epsilon(\phi_L(x,y)) H(\phi_G(x,y) dxdy + \\
&\lambda_3 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H_\epsilon(\phi_L(x,y))(1 - H_\epsilon(\phi_G(x,y)) dxdy.
\end{aligned} \tag{7.9}$$

By keeping ϕ_L and ϕ_G fixed and deriving with respect to c_1 and c_2, we get equations for computing c_1 and c_2:

$$c_1 = \frac{\lambda_{1\mathrm{G}} \int_\Omega u_0 H_\epsilon(\phi_G) dxdy + \lambda_1 \int_\Omega u_0 H_\epsilon(\phi_L) dxdy + \lambda_2 \int_\Omega u_0 (1 - H_\epsilon(\phi_L)) H_\epsilon(\phi_G) dxdy}{\lambda_{1\mathrm{G}} \int_\Omega H_\epsilon(\phi_G) dxdy + \lambda_1 \int_\Omega H_\epsilon(\phi_L) dxdy + \lambda_2 \int_\Omega (1 - H_\epsilon(\phi_L)) H_\epsilon(\phi_G) dxdy}, \tag{7.10}$$

$$c_2 = \frac{\lambda_{2\mathrm{G}} \int_\Omega u_0 (1 - H_\epsilon(\phi_G)) dxdy + \lambda_3 \int_\Omega u_0 (1 - H_\epsilon(\phi_L))(1 - H_\epsilon(\phi_G)) dxdy}{\lambda_{2\mathrm{G}} \int_\Omega (1 - H_\epsilon(\phi_G)) dxdy + \lambda_3 \int_\Omega (1 - H_\epsilon(\phi_L))(1 - H_\epsilon(\phi_G)) dxdy}, \tag{7.11}$$

and by keeping c_1 and c_2 fixed we get the equations for ϕ_G and ϕ_L

$$\begin{cases} \mu_2 g(\nabla u_0)\frac{k}{\gamma^k}\phi_G^{k-1}e^{-(\frac{\phi_G}{\gamma})^k} + \mu_G\nabla\cdot\left((1-\frac{1}{|\nabla\phi_G|})\nabla\phi_G\right)+ \\ \delta_\epsilon(\phi_G)\Big(-\mu_1 d(x,y)g(\nabla u_0)e^{-(\frac{\phi_L}{\gamma})^k} - \lambda_{1\mathrm{G}}(u_0(x,y)-c_1)^2 + \lambda_{2\mathrm{G}}(u_0(x,y)-c_2)^2 - \\ \lambda_2(u_0(x,y)-c_1)^2(1-H(\phi_L)) + \lambda_3(u_0(x,y)-c_2)^2(1-H(\phi_L))\Big) + \\ \alpha g(x,y)|\nabla\phi_G| = 0, \quad \text{in } \Omega \end{cases} \tag{7.12}$$

and

$$\begin{cases} \mu_1 d(x,y)g(\nabla u_0)\frac{k}{\gamma^k}\phi_L^{k-1}e^{-(\frac{\phi_L}{\gamma})^k}H_\epsilon(\phi_G) + \mu_L\nabla\cdot\left((1-\frac{1}{|\nabla\phi_L|})\nabla\phi_L\right)+ \\ \delta_\epsilon(\phi_L)\Big(-\lambda_1(u_0(x,y)-c_1)^2 + \lambda_2(u_0(x,y)-c_1)^2H_\epsilon(\phi_G)+ \\ \qquad \lambda_3(zu_0(x,y)-c_2)^2(1-H_\epsilon(\phi_G))\Big) \ + \ \alpha d(x,y)g(\nabla u_0)|\nabla\phi_L| = 0, \quad \text{in } \Omega \end{cases} \tag{7.13}$$

with $\frac{\partial\phi_G}{\partial\vec{n}} = \frac{\partial\phi_L}{\partial\vec{n}} = 0$ on $\partial\Omega$. The terms $\alpha d(x,y)g(\nabla u_0)|\nabla\phi_L|$ and $\alpha g(x,y)|\nabla\phi_G|$ are the balloon term forces. The approximation can be done by introducing an artificial time step t and getting the gradient descent method. Thus for c_1 and c_2 ,which will be updated at each step according to the above formulas (7.10) and (7.11), we solve

$$\begin{cases} \frac{\partial\phi_G}{\partial t} = \mu_2 g(\nabla u_0)\frac{k}{\gamma^k}\phi_G^{k-1}e^{-(\frac{\phi_G}{\gamma})^k} + \mu_G\nabla\cdot\left((1-\frac{1}{|\nabla\phi_G|})\nabla\phi_G\right)+ \\ \delta_\epsilon(\phi_G)\Big(-\mu_1 d(x,y)g(\nabla u_0)e^{-(\frac{\phi_L}{\gamma})^k} - \lambda_{1\mathrm{G}}(u_0(x,y)-c_1)^2 + \lambda_{2\mathrm{G}}(u_0(x,y)-c_2)^2 - \\ \lambda_2(u_0(x,y)-c_1)^2(1-H(\phi_L)) + \lambda_3(u_0(x,y)-c_2)^2(1-H(\phi_L))\Big) \ + \ \alpha g(x,y)|\nabla\phi_G| = 0 \end{cases} \tag{7.14}$$

and

$$\begin{cases} \frac{\partial\phi_L}{\partial t} = \mu_1 d(x,y)g(\nabla u_0)\frac{k}{\gamma^k}\phi_L^{k-1}e^{-(\frac{\phi_L}{\gamma})^k}H_\epsilon(\phi_G) + \mu_L\nabla\cdot\left((1-\frac{1}{|\nabla\phi_L|})\nabla\phi_L\right)+ \\ \delta_\epsilon(\phi_L)\Big(-\lambda_1(u_0(x,y)-c_1)^2 + \lambda_2(u_0(x,y)-c_1)^2H_\epsilon(\phi_G)+ \\ \qquad \lambda_3(zu_0(x,y)-c_2)^2(1-H_\epsilon(\phi_G))\Big) \ + \ \alpha d(x,y)g(\nabla u_0)|\nabla\phi_L| = 0. \end{cases} \tag{7.15}$$

After solving these equations, the local level set $\phi_L = 0$ will define the selected object.

7.3.2 AOS Algorithm for the Model

In order to develop an AOS method for (7.14) and (7.15), we consider the following related parabolic equations with $\frac{\partial\phi_L}{\partial\vec{n}}\Big|_{\partial\Omega} = \frac{\partial\phi_G}{\partial\vec{n}}\Big|_{\partial\Omega} = 0$:

$$\begin{cases} \frac{\partial\phi_L}{\partial t} = \mu_{\mathrm{L}}\nabla\cdot(E_L\nabla\phi_L) + f_L \ = \mu_{\mathrm{L}}(\partial_x(E_L\partial_x\phi_L) + \partial_y(E_L\partial_y\phi_L)) + f_L, \\ \frac{\partial\phi_G}{\partial t} = \mu_{\mathrm{G}}\nabla\cdot(E_G\nabla\phi_G) + f_G \ = \mu_{\mathrm{G}}(\partial_x(E_G\partial_x\phi_G) + \partial_y(E_G\partial_y\phi_G)) + f_G, \end{cases} \tag{7.16}$$

where

$$
\begin{aligned}
f_L =& d(x,y)g(\nabla u_0)\frac{k}{\gamma^k}\phi_L^{k-1}e^{-(\frac{\phi_L}{\gamma})^k}H_\epsilon(\phi_G) + \delta_\epsilon(\phi_L)\Big(-\lambda_1(u_0(x,y)-c_1)^2 + \\
& \lambda_2(u_0(x,y)-c_1)^2 H_\epsilon(\phi_G) \ + \lambda_3(u_0(x,y)-c_2)^2(1-H_\epsilon(\phi_G))\Big) + d(x,y)g(\nabla u_0)|\nabla\phi_L|, \\
f_G =& \mu g(\nabla u_0)\frac{k}{\gamma^k}\phi_G^{k-1}e^{-(\frac{\phi_G}{\gamma})^k} + \delta_\epsilon(\phi_G)\Big(-\mu_1 d(x,y)g(\nabla u_0)e^{-(\frac{\phi_L}{\gamma})^k} - \lambda_{1\mathrm{G}}(u_0(x,y)-c_1)^2 + \\
& \lambda_{2\mathrm{G}}(u_0(x,y)-c_2)^2 - \lambda_2(u_0(x,y)-c_1)^2(1-H(\phi_L)) + \lambda_3(u_0(x,y)-c_2)^2(1-H(\phi_L))\Big) \\
& + \alpha g(x,y)|\nabla\phi_G|, \\
W =& d(x,y)g(\nabla u_0), \quad E_L = 1-\frac{1}{|\nabla\phi_L|}, \quad E_G = 1-\frac{1}{|\nabla\phi_G|}.
\end{aligned}
$$

It suffices to consider how to solve a general equation:

$$\frac{\partial\phi}{\partial t} = \mu(\partial_x(E\partial_x\phi) + \partial_y(E\partial_y\phi)) + f, \tag{7.17}$$

which by employing the AOS scheme can be split additively as shown in the previous chapters

$$\phi^{n+1} = \frac{1}{m}\sum_{l=1}^{m}(\ I - m\Delta t A_l(\phi^n))^{-1}\ \hat{\phi}^n. \tag{7.18}$$

Details of the tridiagonal matrices A_l for $l = 1, 2$ derived using finite differences have already been given in Chapter 5. The following iteration algorithm has been designed for the model:

Algorithm 7 Infinite Perimeter Dual Level Set Selective Segmentation Model IPDLSSS: $\phi_L^n \leftarrow IPDLSSS(\phi_L^n, u_0, \lambda_1, \lambda_2, \beta, \gamma, k, \Delta t, maxit, tol)$.

Initialization:
Define the markers and associated parameters based on the image being considered,
Design the initial level sets ϕ_L^0 and ϕ_G^0,
Calculate the edge based and distance function according to,
for $n = 1 : maxit$ **do**
Compute $\phi_{L,i,j}^n$ and $\phi_{G,i,j}^n$ using (7.18)
If $\frac{||\phi_L^{n+1}-\phi_L^n||}{||\phi_L^n||} < tol$ set $\phi_{L,i,j}^n \leftarrow \phi_{L,i,j}^{n-1}$, and $\phi_{G,i,j}^n \leftarrow \phi_{G,i,j}^{n-1}$, Break;
$\phi_{L,i,j}^n \leftarrow \phi_{L,i,j}^{n-1}$ and $\phi_{G,i,j}^n \leftarrow \phi_{G,i,j}^{n-1}$,

end for

7.4 Experimental results

To verify the performance of our new method we carried out 3 different sets of experiments on images with oscillating boundaries. The first experimental results demonstrate that our new segmentation method deals properly with the task of selective segmentation. Next, we present the comparison of the new model with the dual level set selective segmentation model in some experimental results. Lastly, we present experimental results of a more challenging problem, where the object shapes have oscillatory boundaries.

For each experiment the markers and some parameters, such as a smoothing factor and the balloon force coefficient, need to be adjusted according to the given image. In our numerical experiments we take: $\triangle t = 0.01$ (for time marching experiments) and $\triangle t = 1$ (for AOS experiments), $h = 1$ (the step space), $k = 8$, $\gamma = 10$, $\alpha = -0.01$, $\tau = 4$, $\mu_{\mathrm{L}} = \mu_{\mathrm{G}} = 10^{-3}$. Different image sizes have been tested $n = 128$, 180, 256, 512. Different values for $\lambda_s = \lambda_{1G} = \lambda_{2G} = \lambda_1 = \lambda_2 = \lambda_3$ have been chosen and show that the method performs well and gets similar results. For the following experiment we will show results of choice for μ_s and λ_s such that:
$\lambda_s = \lambda_{1G} = \lambda_{2G} = \lambda_1 = \lambda_2 = \lambda_3 = 300$, $\mu_s = \mu_1 = \mu_2 = 1$ or
$\lambda_s = \lambda_{1G} = \lambda_{2G} = \lambda_1 = \lambda_2 = \lambda_3 = 300$, $\mu_s = \mu_1 = \mu_2 = 200$ or
$\lambda_s = \lambda_{1G} = \lambda_{2G} = \lambda_1 = \lambda_2 = \lambda_3 = 80$, $\mu_s = \mu_1 = \mu_2 = 1$ or
$\lambda_s = \lambda_{1G} = \lambda_{2G} = \lambda_1 = \lambda_2 = \lambda_3 = 300$, $\mu_s = \mu_1 = \mu_2 = n^2/10$.

Similar to Chapter 5 we chose the ϵ for the Heaviside function equal to 10^{-2} or 1 and the initial global level set a circle

$$\phi_G^0 = \sqrt{(x - x_G^0)^2 + (y - y_G^0)^2} - r_G^0,$$

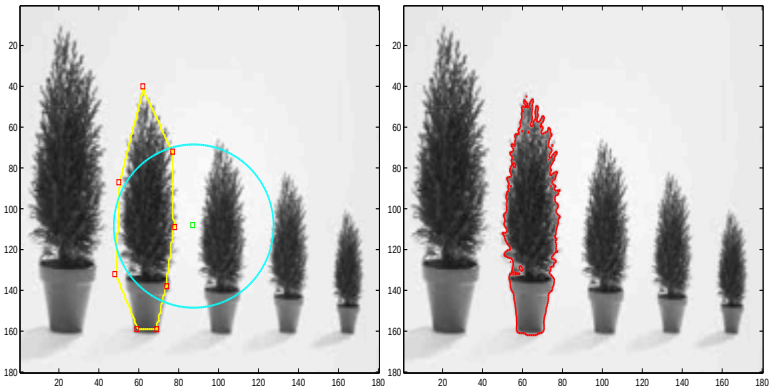
where (x_G^0, y_G^0) is the centre of the circle. The markers can be used in the construction of the local level set. The initial local contour can be a circle such that the centre and the radius are $\mathbf{w_0} = (x_0, y_0) = (\frac{1}{n_1}\sum_{i=1}^{n_1} x_i^*, \frac{1}{n_1}\sum_{i=1}^{n_1} y_i^*)$ and $r = \min_i |\mathbf{w}_i^* - \mathbf{w}_0|$, respectively, where $\mathbf{w}_i^* = (x_i^*, y_i^*)$. In cases when the objects are near (which happens often in such processes), the shape is irregular then we want to start with a better nearby level-set. In this case we can start with a distance function level-set constructed with the polygon of the given markers. This initialization has been found as more proper in our experiments.

7.4.1 Test Set 1 — robustness of the new model

This subsection demonstrates the ability of recognizing specific objects by our infinite perimeter dual level set model by properly segmenting some synthetic and real life images.

First, Figures 7.2 and 7.3 present experimental results using the time marching algorithm for the new dual level set model. The algorithm is based on contracting or expanding the initial curve, therefore the initial curve has to be nearby (inside or outside) the given object. Figure 7.3 shows the local and global level-set initialization. In Figure 7.2(b), we show that the model works satisfactorily with the time marching algorithm for a real life image. Four more different test results have been processed with time marching of which the first two images show that the model works satisfactorily for cases where the features are nearby, while Figure 7.3(c) and (d) shows the successful segmentation of biological and medical images. Next, we show the numerical results of our new method for segmenting 7 different images applying time marching or the AOS method. The images of Figure 7.3 give similar results when processed with the AOS method, and for the sake of brevity we do not show them again. Images in Figure 7.4 show the results obtained using the AOS method. Figure 7.4(b) shows results obtained from the segmentation of images with strong additive noise using the trees test image. Figure 7.4(c) and (d) show the segmentation of a picture of a collection of leaves of different shapes which are close to each other. All of these figures show that by giving some points in the object the model does not get attracted to the other object with

the same intensity.



(a) Initial local (yellow) and global (cyan) level sets (n=180);

(b) Real life image successfully selected(n=180);

Figure 7.2: Successful segmentation with infinite selective segmentation model of the vase with markers set in the boundary of second tree (target object) with time marching algorithm.

7.4.2 Test Set 2 — comparison with the previous Rada-Chen model

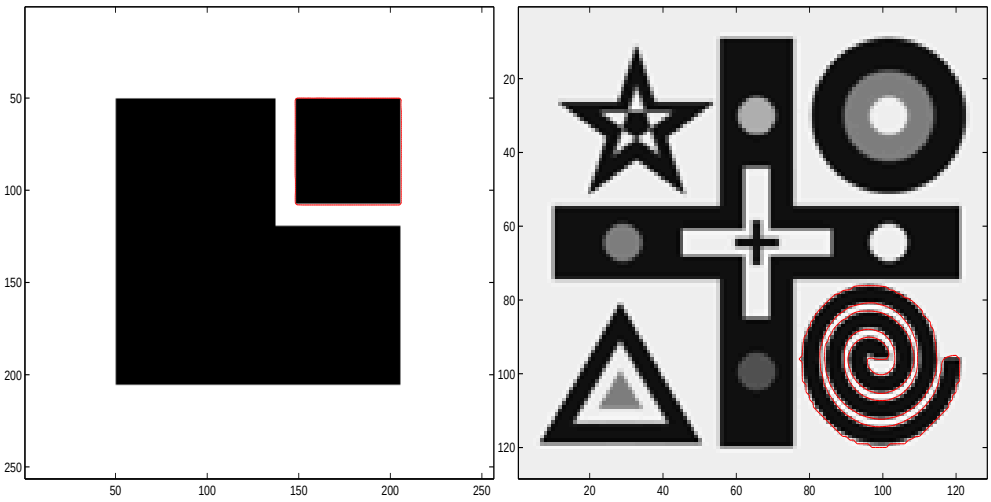
We now compare our model with the dual level set method [129] for 4 easier problems, as shown in Figures 7.5 and 7.6. In these images we can notice that for the test examples, both models give almost the same result except for the fact that the new model is showing the empty place between the leaves much more accurately.

7.4.3 Test Set 3 — improvement of the new model over Rada-Chen model

Test problems 3 are more challenging due to the oscillatory boundaries of the objects. In particular, we consider the problem of segmenting trees since a characteristic of these images is oscillatory boundaries and the cornering effect of the model can be observed. The results of our new segmentation method are shown Figures 7.7, 7.8 and 7.9, which appear to be quite accurate despite the low-quality data. In these figures we show some respective cropping of the images so that the segmentation difference between the images can be easily noticed. In these cases, the previous models from [129] will lose some details. These figures show better results than the infinity dual level set when oscillatory boundaries are present. In each case, the left row image shows the results of [129] and the right image shows the correctly segmented results using our new model.

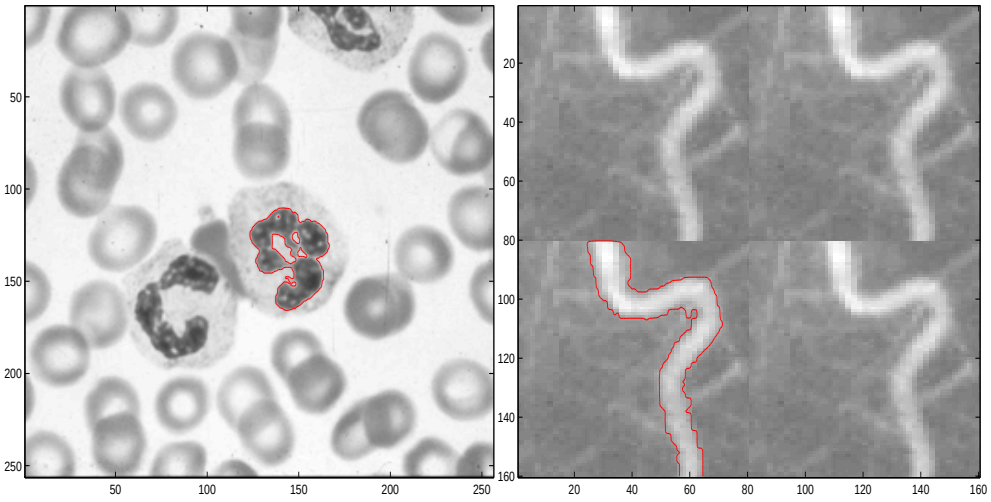
7.5 Conclusions

In this chapter we presented an improved variational model on the previous work of Rada-Chen [129] which selects a target object by using two level set functions (one for global segmentation and the other for local and selective segmentation). To improve and have a more reliable localized segmentation in the case of oscillatory boundaries a Lebesgue measure which leads to no limitation of the perimeter segmentation has been incorporated. We derived the curve evolution equations for the problem posed in the



(a) Geometric image with the markers set in the corner of the rectangle (n=256);

(b) Successful segmentation of the spirals (n=128);

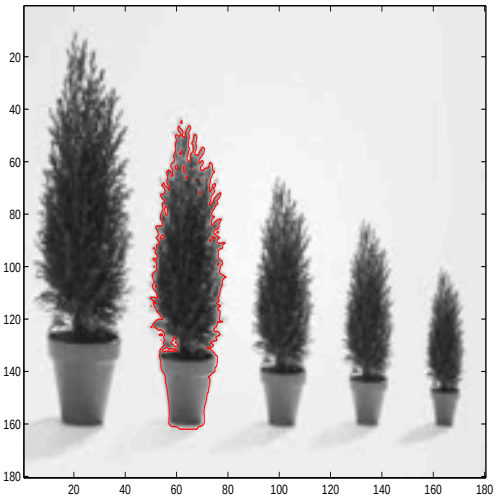


(c) Cells embryo successfully selected (n=256);

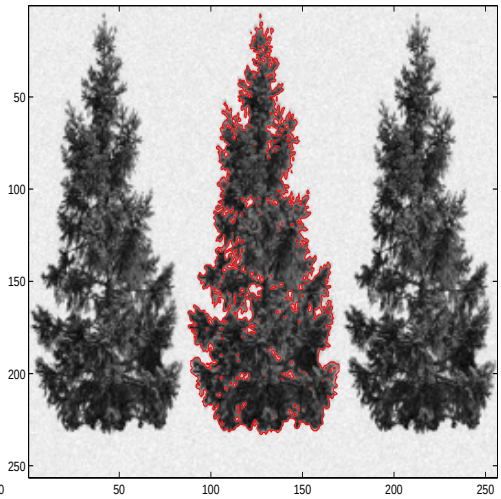
(d) Successful selective segmentation of blood vessel in a repeated image (n=160);

Figure 7.3: Successful segmentation of different images using infinite selective segmentation model with a time marching algorithm.

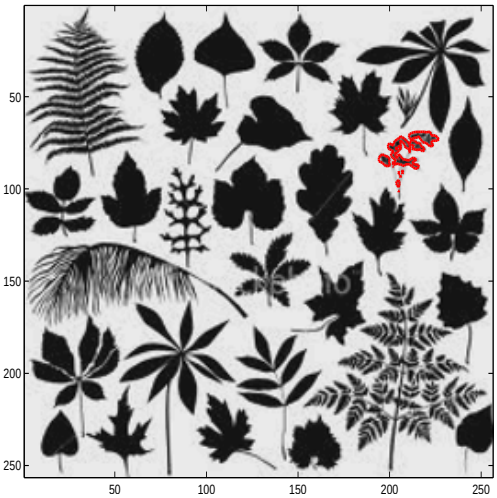
variational framework and demonstrated the effectiveness of the resulting algorithm in segmenting a variety of images. Numerical experiments show that the new model delivers similar results for general problems to old models and improved results for the problems where the boundary has oscillations.



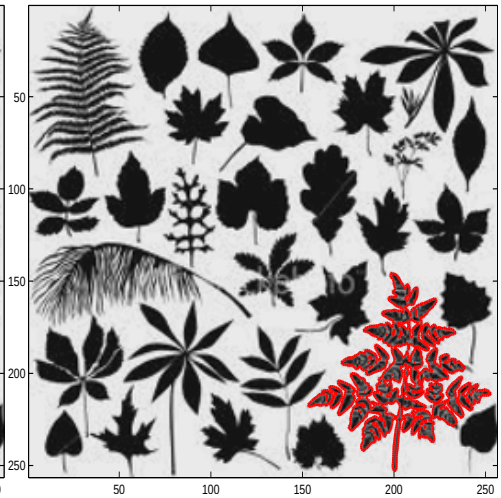
(a) Real life image of vases (n=180);



(b) Real life image of pine trees with 10% additive noise (n=256);



(c) Real life image of collection of leaves (n=256);



(d) Real life image of collection of leaves (n=256);

Figure 7.4: Successful segmentation for different images with oscillatory boundaries with the AOS infinite selective segmentation model, with $dt = 1$, $\mu_s = 1$, $\lambda_s = 300$.

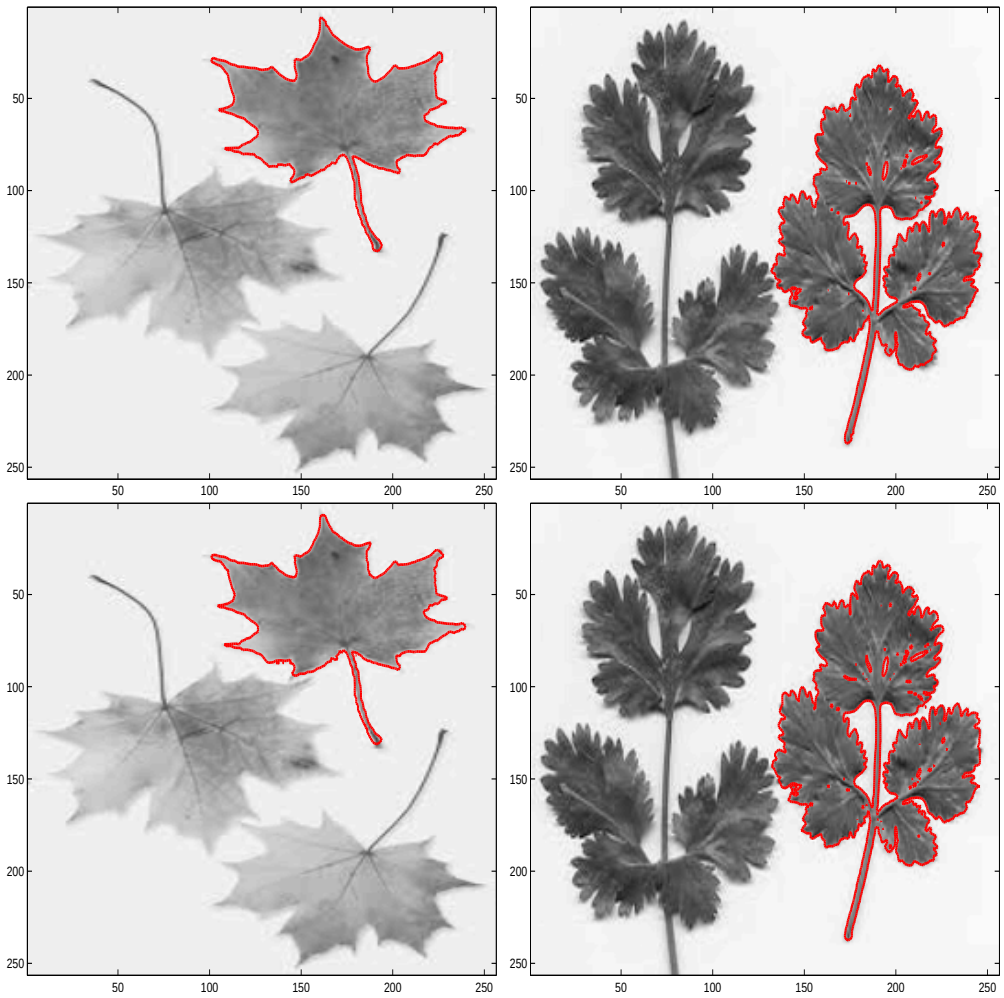


Figure 7.5: Successful segmentation of two tips of leafs with oscillatory boundaries with the AOS infinite selective segmentation model. First row old model with H^1 Hausdorff measurement, second row new model with L^2 measurement, $dt = 1$, $\mu_s = 1$ $\lambda_s = 80$.

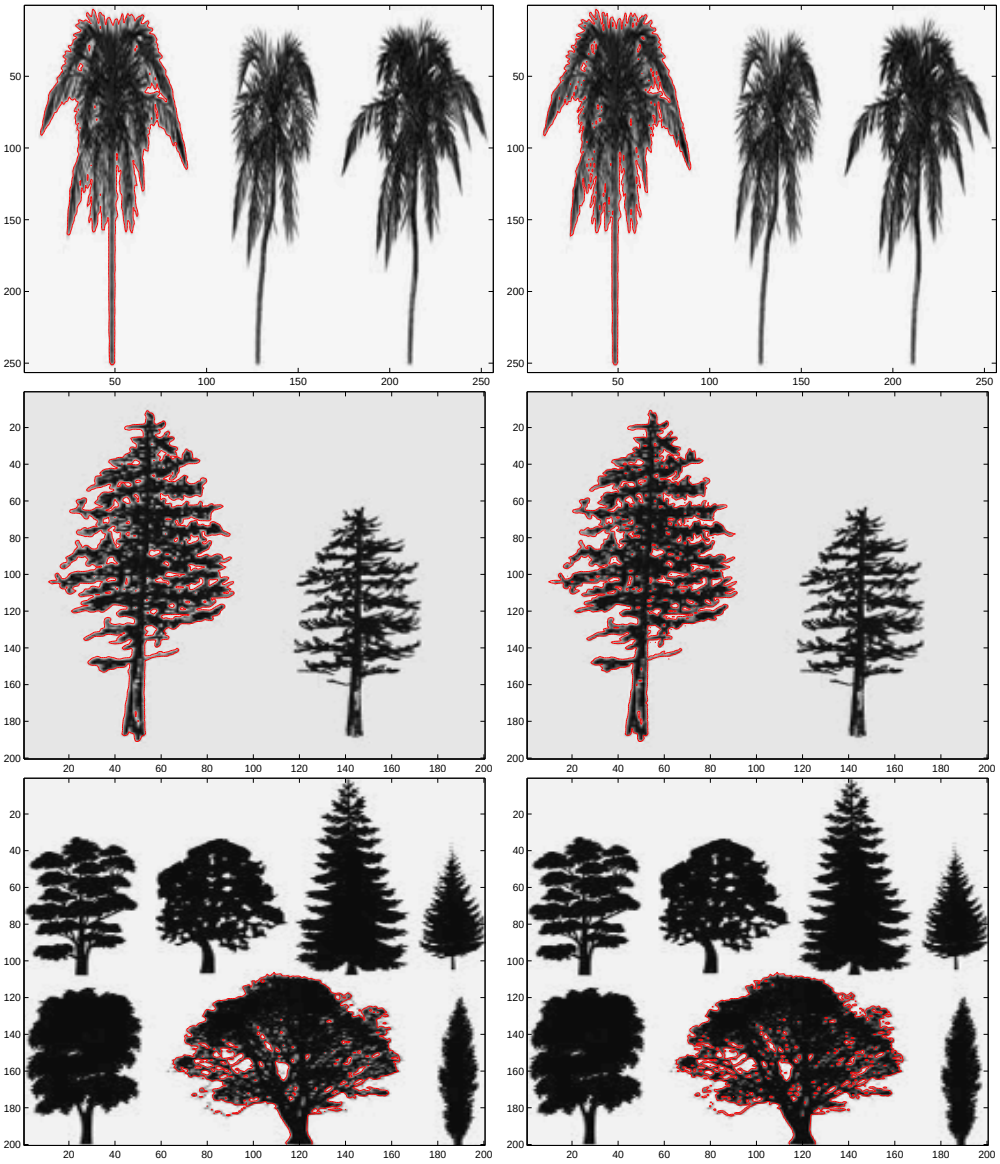


Figure 7.6: Successful segmentation of one of the trees in an image with different trees with the AOS infinite selective segmentation model. The left column shows the segmentation obtained with the old model using H^1 Hausdorff measurement, $dt = 1$, $\mu_s = mn/10$, $\lambda_s = 300$ while the right column shows the segmentation with the new model using L^2 measurement.

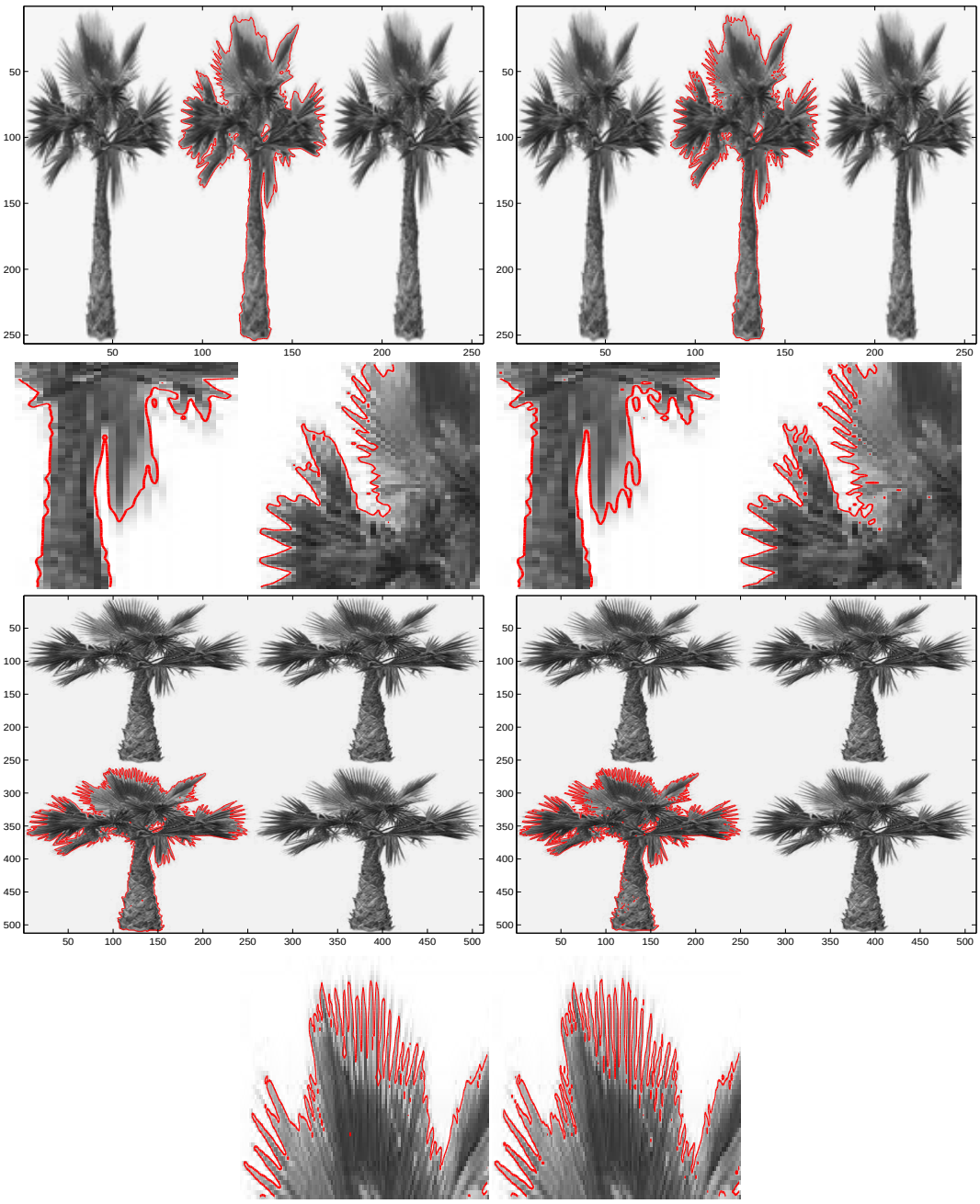


Figure 7.7: Successful selective segmentation of palm trees and respective cropping with the AOS infinite selective segmentation model. First column with old model with H^1 Hausdorff measurement, $dt = 1$, $\mu = mn/10$ $\lambda = 300$, second column new model with L^2 measurement

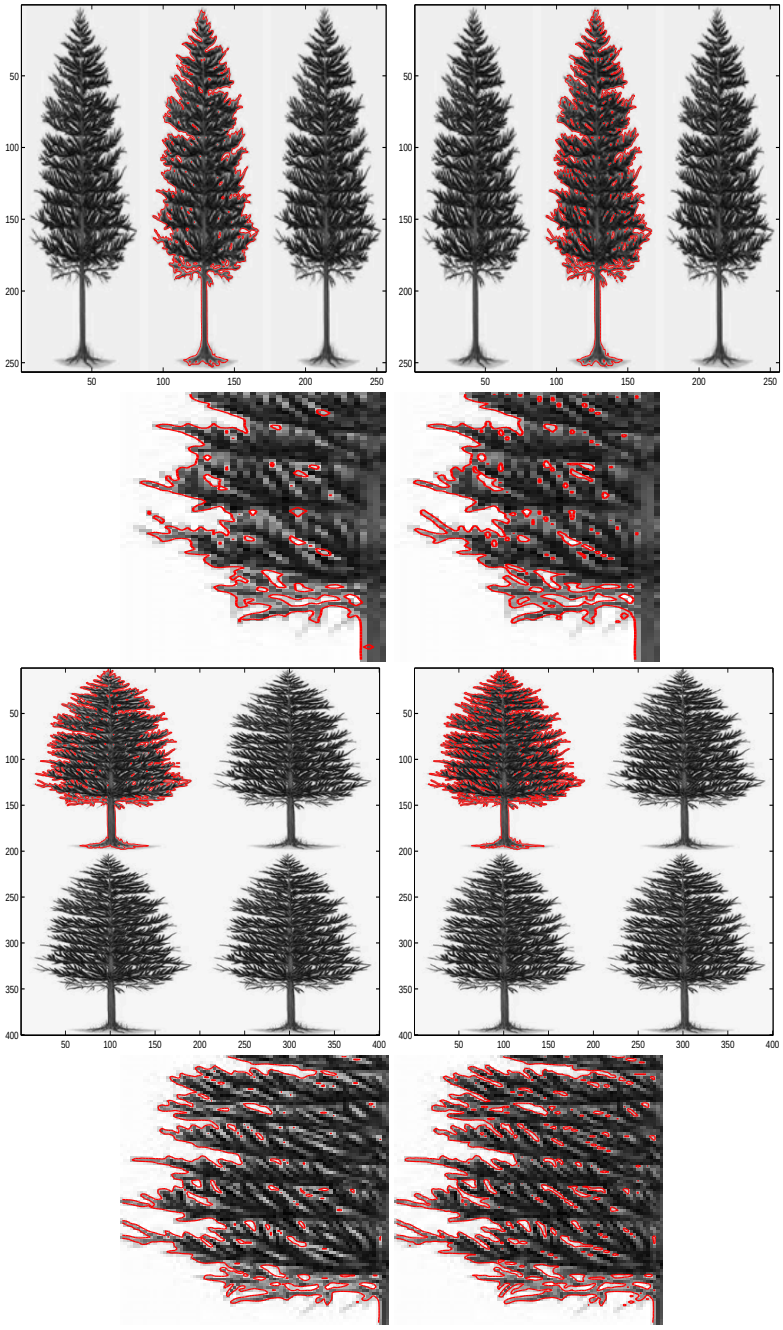


Figure 7.8: Successful selective segmentation of one of the pine trees and the respective cropping with the AOS infinite selective segmentation model. First column with old model with H^1 Hausdorff measurement, $dt = 1$, $\mu = mn/10$ $\lambda = 300$, second column new model with L^2 measurement.

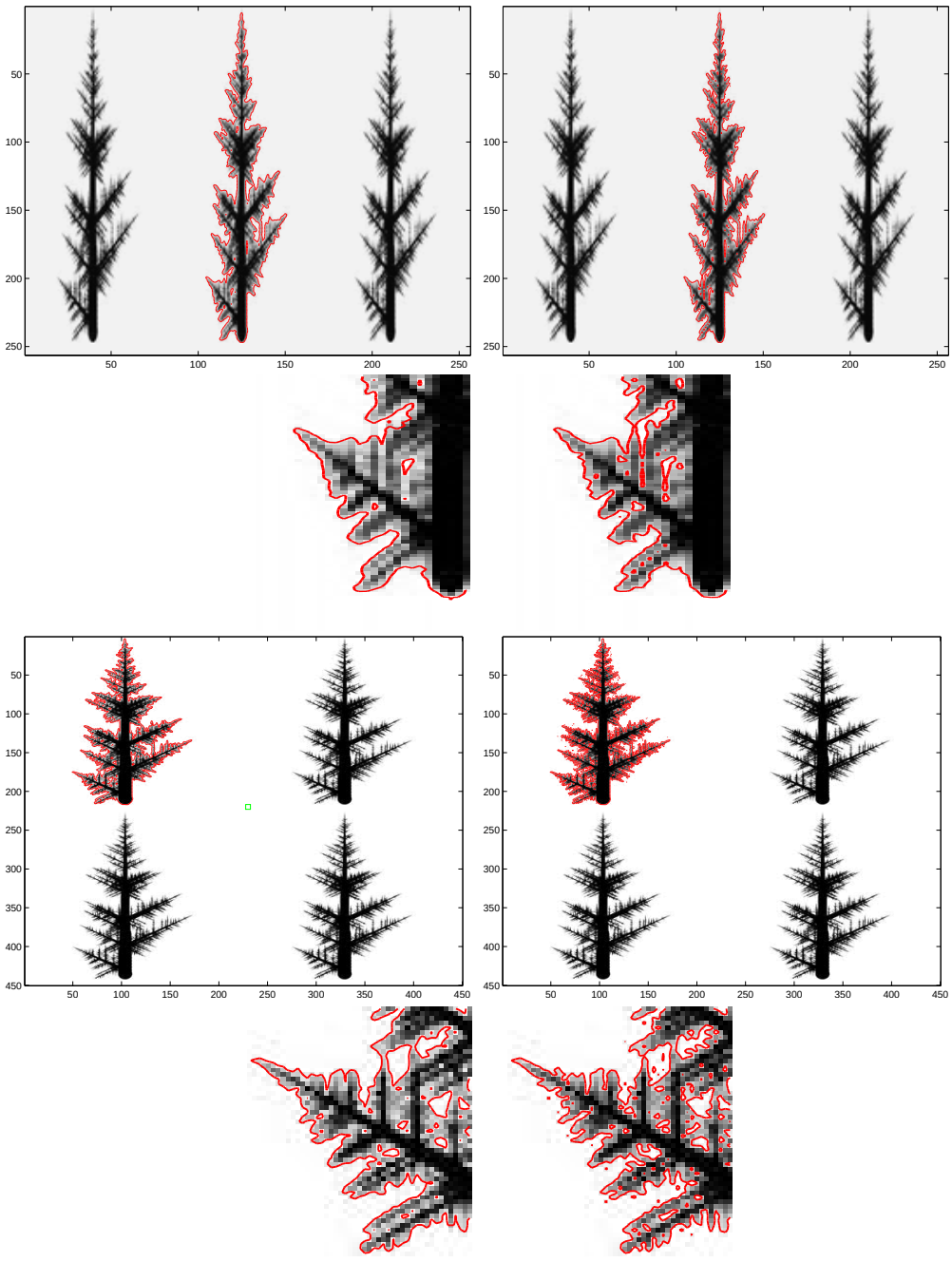


Figure 7.9: Successful selective segmentation of pine branch and the respective cropping with the AOS infinite selective segmentation model. First column with old model with H^1 Hausdorff measurement, $dt = 1$, $\mu = mn/10$ $\lambda = 300$, second column new model with L^2 measurement.

Chapter 8

Improved Selective Segmentation Model Using One Level-Set and Its Viscosity Solution

The main aim of this chapter is the construction of an efficient variational model for selective segmentation tasks where only one object is required to be extracted automatically. As mentioned in the previous chapters this kind of technique is of crucial interest in many fields, as an automation of these tasks will reduce time spent by field workers doing this manually.

In this chapter we propose a variational single level-set selective segmentation based model which is much faster to implement than the previous most efficient model by Rada-Chen [129], detailed in Chapter 5. The model incorporates several new tools which brings us to a new model with the same efficiency and reliability when compared to Rada-Chen [129] meanwhile being faster. The model will be compared with latest interactive image segmentation algorithms, such as Nguyen-Cai-Zhang-Zheng method [8], showing the same performance or improvement to segmentation in case of object with same intensity. On the other hand, we provide an answer to the existence and uniqueness of the solution associated to our problem by using viscosity theory. Numerical results are given through an AOS algorithm for solving the resulting Euler-Lagrange equation. The experimental results show that the model finds the desired local boundaries successfully in various challenging cases and it is not heavily dependent on the prior information of markers or the distance function which is based on them. More importantly, the new model gives an overall improvement over the previous models and can be recommended for selective segmentation.

8.1 Introduction

In order to separate objects from their surroundings many different techniques have been developed so far [107, 146, 167, 4, 188, 10, 28, 149, 142, 39, 83, 71, 109, 162, 37, 115, 122, 34, 39, 166] etc., and are proven to be very efficient for global segmentation where all features in an image are required to be segmented.

Selective segmentation is a new subject developed in recent years and not yet been widely studied. As mentioned in the previous chapters there have been a few variational selective segmentation methods [66, 67, 68, 133, 69, 12, 129, 8]. Shortly, Gout-Guyader-Vese [66, 67, 69] proposed an edge based method for selective segmentation and it

was improved by a mixed edge-based and region-based model by Badshah-Chen [12]. Other methods such as Random Walks [68], Geodesic [13] and GrabCut [133] uses the distributions probability, edge based function or graph cut theory and the authors of [8] improved these models by introduced an interactive image segmentation algorithm less sensitive to the user inputs and sensitivity to small variations able to produce an accurate boundary with a small amount of user interaction. The recent work by Rada-Chen [129] has shown to give satisfactory results for cases where the other methods would produce spurious objects (i.e. fail the selection) in some hard cases where the objects are near to each other or the intensity difference is small, as shown by the example given in Fig. 5.2.

The drawback of the Rada-Chen model [129] is that it is slow to implement due to the update of two level set functions in comparison with the previous single level set function. This brings us back to the idea of designing a model which uses only a single level set function with the requirement of having the same reliability as the dual level set [129] and will ensure same or better performance compared with [8]. To solve this problem we are going to introduce a new method which is based on one level set function, which will ensure a better performance than all previous *one level set* methods of this type while having the same results as the present method which uses dual level sets. The main issues for any new method would be to provide i) ability to extract smooth regular contours of the target object; ii) robustness of the algorithms: stability, convergence with a minimum set of parameters; iii) theoretical correctness of the method.

The new method employs i) a crucial new area-based minimization fitting term considered to enhance the model's reliability (such as an area fitting term which serves as a constraint rather than precise area preservation); ii) an adaptive parameter edge detection function to better influence and decrease the functional as soon as we are in the boundaries; iii) less significance of the distance function from the given geometrical points (using it only when it is feasible to give an accurate estimate of the object of interest); iv) a new region-based selective segmentation fitting term which optionally updates the mean intensity of the background. We aim to provide an answer to the existence and uniqueness of the solution associated with our parabolic problem using the theory of viscosity. We will consider two similar minimization problems with the difference that in one of them the region fitting term does not update the mean intensity of the background and the second one where this term has been updated. Numerical results will be presented as well to compare the new model with other models. It will be shown that the new model gives as satisfactory results as the Rada-Chen model [129].

This chapter is organized in the following way. In Section §8.2 we present our new model of minimization and derive the Euler-Lagrange equations. In Section §8.3 we first give some general background followed by an extension of Ishii-Sato [78] and Gout *et al.* [65] viscosity solution works to our model problem. Other viscosity solutions of nonlinear elliptic and parabolic PDEs, with nonlinear Neumann boundary conditions, such as those given by Barles [16, 15], Alvarez *et al.* [6] or Caselles [27] can be given but which with the same argument details in ref. [65] Ishii-Sato [78] is more proper. In this approach the conditions are less restrictive and the theorem is more appropriate to our considered problem. The extension of the existence and uniqueness for the second minimization approach requires some appropriate restriction while we have to acknowledge that in the numerical experiments this model finds the boundaries faster. In Section §8.4 we describe the discretization of the method and develop an additive

operator splitting algorithm for solving the PDE, which is very efficient for this kind of problem. In Section §8.5 we give some experimental results. Examples of application of the method are presented to different data-sets and along with comparison to Badshah-Chen [12] and the dual level set [129] selective segmentation algorithms. Conclusions are given in Section §8.6.

8.2 A New One Level Selective Segmentation Variational Model

In this section we propose a novel one level selective segmentation variational model, using the key idea of area fitting. Similar to Chapter 5 we aim to detect an image feature/object that is close to the geometrical domain of the points $\mathcal{A}$ which are placed inside the object or in the boundaries.

The geometrical domain of the points $\mathcal{A}$ is a good indicator of where our initialization for the level set has to be placed and at the same time an indicator of the relative size of the object in the area term. We can get an approximate ratio of the object in comparison with the whole domain Ω considering the polygon area constricted with the given points. On the other hand the mean intensity of the polygon into the points is approximately the intensity of the target object, which can be one more piece of information to be used. The area outside of this polygon can also be as well an indicator of the mean intensity outside the target object, which can be as well updated while the curve evolves. This brings us to two cases which will be considered in this section. Even though there is no considerable difference in the experimental results, the non updated case is much more convenient in terms of proving existence and uniqueness of the theory for the model.

Recalling from Chapters 2 and 5, the distance function gives a local weight in a small neighborhood of given markers. In order to have good influence from this function we need to increase the number of points in the set $\mathcal{A}$ and for more they have to be near the boundaries, which is not practically convenient. For this reason we want the program to be less dependent of the distance function and only needed when it is feasible to give an accurate estimate of the object of interest. In this way we do not fail in cases where inaccurate geometrical information is given.

Considering all of the given arguments we propose the following energy minimization functional:

$$\begin{aligned}\min_{\Gamma,c_2} F(\Gamma,c_2) = \Big\{ \mu \int_{\Gamma} g(|\nabla u_0(x,y)|)dxdy + \quad & \lambda_1 \int_{inside(\Gamma)} |u_0(x,y)-c_1|^2 dxdy + \\ & \lambda_2 \int_{outsite(\Gamma)} |u_0(x,y)-c_2|^2 dxdy + \\ & \nu\Big(\int_{inside(\Gamma)} d\xi d\eta - A_1 \Big)^2 + \Big(\int_{outsite(\Gamma)} d\xi d\eta - A_2 \Big)^2 \Big\},\end{aligned} \tag{8.1}$$

where $\lambda_1, \lambda_2, \mu, \nu$ are empirical weights, c_1 is the known mean of the polygon constructed with the given markers (with the assumption that the markers are placed inside the object or outside and not too far from the boundaries), c_2 is a region term which is considered as the mean intensity of the domain outside the object. The term c_2 can either computed as a mean of the domain left out of the polygon constructed with the given markers (which will have an approximated error depending on the initial

markers) or we may update it as the level set evolves, defined below in equation (8.6) representing the mean intensity outside the target object. This will distinguish two models named *Model 1* and *Model 2*, respectively. As an edge detection function we consider an adaptive parameter function g given by:

$$g(|\nabla u_0(x,y)|) = \frac{1}{1+k|\nabla u_0(x,y)|^2}, \tag{8.2}$$

with k a positive constant. The constant k helps to sharpen the edges especially when the intensity difference is small, since in this case we know that near the edges $|\nabla u_0(x,y)|$ might not be big enough such that the function $g(|\nabla u_0(x,y)|)$ is approximately 0. A Gaussian convolution of the image $u_0(x,y)$ with $G_\sigma(x,y) = \sigma^{-1/2}e^{-|x^2+y^2|/4\sigma}$, can be used in cases of strong noise, e.g., $G_\sigma(x,y) * u_0(x,y)$, which helps to eliminate the height of non-desired frequencies.

Analyzing the energy terms in (8.1), the first term of the given energy above (weighted by μ) is a weighted geodesic length of the contour. The second and the third term (weighted by λ_1, λ_2) are region fitting terms to the mean intensity inside and outside the object, respectively. The forth and fifth term (weighted by ν) are a priori terms stating that the volume area of each object remains close to a reference area (or volume for 3-D) A_i, $i = 1, 2$. The constants A_i are the area of the polygon inside and outside the given markers.

As underlined in the beginning of the chapter our model does not contain the distance function (unlike [12, 129]) unless the boundaries are feasible to give an accurate estimate of the object of interest. Rewriting the above equation in terms of the level-set function we have:

$$\begin{aligned}\min_{\phi(x,y),c_2} F(\phi(x,y),c_2) = \Big\{\mu \int_\Omega g(|\nabla u_0(x,y)|)|\nabla H(\phi(x,y))|dxdy + \\ \lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H(\phi(x,y))dxdy + \\ \lambda_2 \int_\Omega |u_0(x,y)-c_2|^2 (1-H(\phi(x,y)))dxdy + \\ \nu\Big(\int_\Omega H\left(\phi(\xi,\eta)\right)d\xi d\eta - A_1\Big)^2 + \\ \nu\Big(\int_\Omega \left(1-H(\phi(\xi,\eta)\right)d\xi d\eta - A_2\Big)^2\Big\}.\end{aligned} \tag{8.3}$$

Replacing the non-differentiable H function by H_ϵ, a regularized Heaviside function as in [10, 37], we obtain our *Model 2*:

$$\begin{aligned}\min_{\phi(x,y),c_2} F_\epsilon(\phi(x,y),c_2) = \Big\{\mu \int_\Omega g(|\nabla u_0(x,y)|)\delta_\epsilon(\phi(x,y))|\nabla(\phi(x,y))|dxdy + \\ \lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H_\epsilon(\phi(x,y))dxdy + \\ \lambda_2 \int_\Omega |u_0(x,y)-c_2|^2 (1-H_\epsilon(\phi(x,y)))dxdy + \\ \nu\Big[\Big(\int_\Omega H_\epsilon(\phi(\xi,\eta))d\xi d\eta - A_1\Big)^2 + \Big(\int_\Omega (1-H_\epsilon(\phi(\xi,\eta))d\xi d\eta - A_2\Big)^2\Big]\Big\},\end{aligned} \tag{8.4}$$

where $\delta_\epsilon(\phi(x,y))$ is a regularized Delta function corresponding to the Heaviside function introduced above. Here by *Model 2* i.e. (8.4), we mean the general case with two unknowns ϕ and c_2. We shall consider *Model 1* the case which does not update the background mean intensity value c_2 while the curve evolves toward the object by keeping it as a constant equal to the mean intensity of the background of the first initial level set, shortly:

$$\begin{aligned}\min_{\phi(x,y)} F_\epsilon(\phi(x,y)) = \Big\{\mu \int_\Omega g(|\nabla u_0(x,y)|)\delta_\epsilon(\phi(x,y))|\nabla(\phi(x,y))|dxdy + \\ \lambda_1 \int_\Omega |u_0(x,y)-c_1|^2 H_\epsilon(\phi(x,y))dxdy + \\ \lambda_2 \int_\Omega |u_0(x,y)-c_2|^2 (1-H_\epsilon(\phi(x,y)))dxdy + \\ \nu\Big[\Big(\int_\Omega H_\epsilon(\phi(\xi,\eta))d\xi d\eta - A_1\Big)^2 + \Big(\int_\Omega (1-H_\epsilon(\phi(\xi,\eta))d\xi d\eta - A_2\Big)^2\Big]\Big\},\end{aligned} \tag{8.5}$$

where both c_1 and c_2 are estimated from the given polygon P.

Keeping $\phi(x,y)$ fixed back to equation (8.4) and minimizing with respect to c_2, the unknown intensity outside the object, one gets the following equality for computing c_2:

$$c_2(\phi(x,y)) = \frac{\int_\Omega u_0(x,y)(1-H_\epsilon(\phi(x,y)))dxdy}{\int_\Omega (1-H_\epsilon(\phi(x,y)))dxdy} \tag{8.6}$$

if $\int_\Omega (1-H_\epsilon(\phi(x,y)))dxdy > 0$ (i.e if the curve is nonempty in Ω).

Keeping c_1 and c_2 fixed, we minimize (8.4) with respect to $\phi(x,y)$ to derive briefly the Euler-Lagrange equation. We recall that $\mathcal{F}$ is differentiable in the Gâteaux sense at $\phi \in X$ if the limit

$$\mathcal{F}'_\phi(\psi) = \frac{d}{dh}\Big(\mathcal{F}(\phi+h\psi)\Big)\Big|_{h=0} = \lim_{h\to 0}\frac{\mathcal{F}(\phi+h\psi)-\mathcal{F}(\phi)}{h},$$

is defined for any $\psi \in X$. Coming back to our problem, let us determine the Gâteaux derivative of the energy F_ϵ and find the first variation of the functional F_ϵ with respect to ϕ such that:

$$\lim_{h\to 0}\frac{d}{dh}\Big(\mathcal{F}_\epsilon(\phi+h\psi, c_2)\Big) = 0.$$

Using the notation ϕ, u_0 instead of $\phi(x,y)$, $u_0(x,y)$ we have

$$\begin{aligned}&\mu\frac{d}{dh}\int_\Omega g(|\nabla u_0|)\delta_\epsilon(\phi+h\psi)|\nabla(\phi+h\psi)|dxdy\Big|_{h=0} + \\ &\frac{d}{dh}\int_\Omega \Big(\lambda_1|u_0-c_1|^2 H_\epsilon(\phi+h\psi) + \lambda_2|u_0-c_2|^2(1-H_\epsilon(\phi+h\psi)) + \\ &\nu\Big\{(\int_\Omega H_\epsilon(\phi+h\psi)d\xi d\eta - A_1)^2 + (\int_\Omega (1-H_\epsilon(\phi+h\psi))d\xi d\eta - A_2)^2\Big\}\Big|_{h=0} = 0\end{aligned}$$

$$\Longrightarrow$$

$$\mu\int_\Omega g(|\nabla u_0|)\Big(|\nabla(\phi+h\psi)|\frac{d}{dh}\delta_\epsilon(\phi+h\psi)dxdy+\delta_\epsilon(\phi+h\psi)\frac{d}{dh}|\nabla(\phi+h\psi)|\Big)dxdy\Big|_{h=0}+$$
$$\int_\Omega\Big(\lambda_1(u_0(x,y)-c_1)^2\delta_\varepsilon(\phi+h\psi)\psi\ -\lambda_2(u_0(x,y)-c_2)^2\delta_\varepsilon(\phi+h\psi)\psi\Big)dxdy+$$
$$2\nu\int_\Omega\Big\{\Big(\int_\Omega H_\varepsilon(\phi+h\psi)d\xi d\eta-A_1\Big)\delta_\varepsilon(\phi+h\psi)\psi-$$
$$\Big(\int_\Omega(1-H_\varepsilon(\phi+h\psi))d\xi d\eta-A_2\Big)\delta_\varepsilon(\phi+h\psi)\psi dxdy\Big\}\Big|_{h=0}=0.$$

Scaling ν and computing the derivatives we have:

$$\mu\int_\Omega g(|\nabla u_0|)\Big(\delta'_\epsilon(\phi)|\nabla\phi|\psi+\delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi\Big)dxdy+$$
$$\int_\Omega\delta_\epsilon(\phi)\Big(\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big)\psi dxdy$$
$$\nu\int_\Omega\delta_\epsilon(\phi)\Big\{(\int_\Omega H_\varepsilon(\phi)d\xi d\eta-A_1)-(\int_\Omega(1-H_\varepsilon(\phi))d\xi d\eta-A_2)\Big\}\psi dxdy=0$$

or

$$\mu\int_\Omega g(|\nabla u_0|)\delta'_\epsilon(\phi)|\nabla\phi|\psi dxdy+\mu\int_\Omega g(|\nabla u_0|)\delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi dxdy+$$
$$\int_\Omega\delta_\epsilon(\phi)\Big(\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big)\psi dxdy+ \tag{8.7}$$
$$\nu\int_\Omega\delta_\epsilon(\phi)\Big\{(\int_\Omega H_\varepsilon(\phi)d\xi d\eta-A_1)-(\int_\Omega(1-H_\varepsilon(\phi))d\xi d\eta-A_2)\Big\}\psi dxdy=0$$

where ψ is a test function of the same type as ϕ. Applying Green's identity

$$\int_\Omega v\nabla\cdot\vec{w}dx=-\int_\Omega\nabla v\cdot\vec{w}dx+\int_{\partial\Omega}v\vec{w}\cdot\vec{n}ds$$

to the second integral of (8.7) by taking $\psi=v$ and $\vec{w}=g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi$ for the second integral we rewrite them respectively as

$$\int_\Omega g(|\nabla u_0|)\delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi dx=-\int_\Omega\nabla\cdot\Big(g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\nabla\phi\Big)\psi dxdy+$$
$$\int_{\partial\Omega}g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}}\psi ds,$$

where $\nabla\phi\cdot\vec{n}=\frac{\partial\phi}{\partial\vec{n}}$. Thus equation (8.7) becomes

$$\begin{cases}\mu\int_\Omega g(|\nabla u_0|)\delta'_\epsilon(\phi)|\nabla\phi|\psi dxdy\ +\mu\int_{\partial\Omega} g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}}\psi ds\\ -\mu\int_\Omega\nabla\cdot\Big(\delta_\epsilon(\phi)g(|\nabla u_0|)\Big)\frac{\nabla\phi}{|\nabla\phi|}\Big)\psi dxdy\ +\\ \int_\Omega\delta_\epsilon(\phi)\Big(\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big)\psi dxdy+\\ \nu\int_\Omega\delta_\epsilon(\phi)\Big((\int_\Omega H_\varepsilon(\phi)d\xi d\eta-A_1)-(\int_\Omega(1-H_\varepsilon(\phi))d\xi d\eta-A_2)\Big)\psi dxdy=0,\end{cases}$$

$$\Longrightarrow\quad\begin{cases}\mu\int_\Omega g(|\nabla u_0|)\delta'_\epsilon(\phi)|\nabla\phi|\psi dxdy\ +\ \mu\int_{\partial\Omega} g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}}\psi ds\ -\\ \mu\int_\Omega\delta_\epsilon(\phi)\nabla\cdot\Big(g(|\nabla u_0|)\frac{\nabla\phi}{|\nabla\phi|}\Big)\psi dxdy\ -\ \mu\int_\Omega\delta'_\epsilon(\phi_L)g(|\nabla u_0|)\nabla\phi\cdot\frac{\nabla\phi}{|\nabla\phi|}\psi dxdy+\\ \int_\Omega\delta_\epsilon(\phi)\Big(\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big)\psi dxdy\\ \nu\int_\Omega\delta_\epsilon(\phi)\Big((\int_\Omega H_\varepsilon(\phi)d\xi d\eta-A_1)-(\int_\Omega(1-H_\varepsilon(\phi))d\xi d\eta-A_2)\Big)\psi dxdy=0.\end{cases}$$

This gives

$$\begin{cases}-\mu\int_\Omega\delta_\epsilon(\phi)\nabla\cdot\Big(g(|\nabla u_0|)\frac{\nabla\phi}{|\nabla\phi|}\Big)\psi dxdy\ +\ \mu\int_{\partial\Omega} g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}}\psi ds\ +\\ \int_\Omega\delta_\epsilon(\phi)\Big(\lambda_1(z-c_1)^2-\lambda_2(z-c_2)^2\Big)\psi dxdy\ +\\ \nu\int_\Omega\delta_\epsilon(\phi)\Big[(\int_\Omega H_\varepsilon(\phi)d\xi d\eta-A_1)-(\int_\Omega(1-H_\varepsilon(\phi))d\xi d\eta-A_2)\Big]\psi dxdy=0.\end{cases}$$

With the notation $W=g(|\nabla u_0|)$ the following Euler-Lagrange equation for ϕ can be derived:

$$\begin{cases}\mu\delta_\epsilon(\phi)\nabla\cdot\Big(W\frac{\nabla\phi}{|\nabla\phi|}\Big)\ +\ \delta_\epsilon(\phi)\Big\{-\lambda_1(u_0(x,y)-c_1)^2+\lambda_2(u_0(x,y)-c_2)^2\ -\\ \nu\Big[(\int_\Omega H_\varepsilon(\phi)dxdy-A_1)-(\int_\Omega(1-H_\varepsilon(\phi))dxdy-A_2)\Big]\Big\}=0\quad\text{ in }\Omega\end{cases}\tag{8.8}$$

where the boundary condition $\mu g(|\nabla u_0|)\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}}=0$ reduces to the Neumann boundary condition. Equation (8.8) can be rewritten as:

$$\begin{aligned}\delta_\epsilon(\phi)\Big\{\mu\nabla\cdot\Big(W\frac{\nabla\phi}{|\nabla\phi|}\Big)-\Big[\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big]-\\ \nu\Big[\Big(\int_\Omega H dxdy-A_1\Big)-\Big(\int_\Omega(1-H)dxdy-A_2\Big)\Big]\Big\}=0,\quad\text{ in }\Omega\end{aligned}\tag{8.9}$$

with $\frac{\partial\phi}{\partial\vec{n}}\Big|_{\partial\Omega}=0$.

In case the markers are near the boundaries $W=g(|\nabla u_0(x,y)|)$ can be replaced by $W=d(x,y)g(|\nabla u_0(x,y)|)$, similar to Badshah-Chen or Gout-Guyader [12, 69]. In equation (8.9) balloon terms such as $\alpha W|\nabla\phi|$ can be added to speed up the convergence.

The final equations of ϕ can be written in the form:

$$\begin{aligned}\delta_\epsilon(\phi)\Big\{\mu\nabla\cdot\Big(W\frac{\nabla\phi}{|\nabla\phi|}\Big)-\Big[\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big]-\\ \nu\Big[\Big(\int_\Omega H dxdy-A_1\Big)-\Big(\int_\Omega(1-H)dxdy-A_2\Big)\Big]\Big\}-\alpha W|\nabla\phi|=0.\end{aligned} \tag{8.10}$$

As in the previous chapters, to avoid re-initialization of the level set function $\phi(x,y)$ we employ a similar term to the Li-Xu-Gui-Fox idea [92] of the form

$$\theta\nabla\cdot\Big((1-\frac{1}{|\nabla\phi|})\nabla\phi\Big).$$

For more about re-initialization techniques refer to [92, 121]. In our experiments re-initialization was not required.

The approximation of equation (8.10) can be done by introducing an artificial time step t and getting the gradient descent method. In this way we get the following evolution equation:

$$\begin{aligned}\frac{\partial\phi}{\partial t}=\delta_\epsilon(\phi)\Big\{\mu\nabla\cdot\Big(W\frac{\nabla\phi}{|\nabla\phi|_\beta}\Big)-\Big[\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big]-\\ \nu\Big[\Big(\int_\Omega H dxdy-A_1\Big)-\Big(\int_\Omega(1-H)dxdy-A_2\Big)\Big]\Big\}-\alpha W|\nabla\phi|=0,\end{aligned} \tag{8.11}$$

with Neumann boundary conditions and β a small parameter which helps to avoid singularities of the term $|\nabla\phi|$ by replacing it with $|\nabla\phi|_\beta=\sqrt{\phi_x^2+\phi_y^2+\beta}$.

8.3 Existence and Uniqueness Based on Viscosity Solution

8.3.1 General Viscosity Background

In this section we discuss the wellposedness in the mathematical sense of the geometric PDE model, that is, we prove existence and uniqueness of solutions in the viscosity sense for Neumann boundary continuous. Both models will be considered, *Model 1* with a fixed c_2 and *Model 2* where c_2 has been updated. Both cases bring us to a singular parabolic equations with Neumann boundary conditions.

Similar types of problems have already been considered by different authors [6, 15, 16, 17, 48, 62, 78, 103]. Our main motivation is to extend the results of Caselles [27] and Gout [66, 65] into our proposed model. For a deep understanding the viscosity solution we refer the reader to the "User's guide to viscosity solutions" of Crandall, Ishii and Lions [48] and to the paper of Gout-Guyader [65].

We only recall here some notion of viscosity solution for a given function

$$F(t,x,\phi,D\phi,D^2\phi)=0,\quad\text{in }\Omega,$$

or similarly for

$$\phi_t+F(t,x,\phi,D\phi,D^2\phi)=0,\ \text{in }[0,T]\times\Omega,$$

with Neumann boundary conditions denoted

$$B(x, D\phi) = 0, \quad \text{on } \partial\Omega,$$

where B is a real-valued continuous function.

The classical concept of what is meant by a "solution" [1] to a partial differential equation (PDE) is generalized by Pierre-Louis Lions [103] and Michael Crandall [48].

We use here the existence theorem and the same notion of viscosity solutions with the "User's guide to viscosity solutions" by Crandall et al. [48] and Gout paper [65] based on the Ishii and Sato [78] work. This theory applies to some PDEs that can formally be written in the form $F(x, u, Du, D^2u)$ where Du denotes the gradient and D^2u the Hessian matrix (which is symmetric).

Let F be defined as $F : R^n \times R \times \ R^n \times \ S(n) \to R$ where $S(n)$ denotes the set of symmetric matrices. In our case, F will be defined by $F : \Omega \times R \times \ R^n \times \ S(n) \to R$. As general background in which the viscosity theory can be introduced is the so called *F being proper*. Two conditions (monotonicity conditions) are necessary to apply this theory to an equation of the type $F = 0$, namely:

$$\begin{array}{ll} (\mathbf{P1}) & F(x,r,p,X) \leq F(x,s,p,X) \text{ with } r \leq s. \\ (\mathbf{P2}) & F(x,r,p,Y) \leq F(x,r,p,X) \text{ with } X \leq Y. \end{array} \tag{8.13}$$

The condition (**P2**) is called degenerate ellipticity. When both conditions (**P1**, **P2**) hold, F is said to be proper. Note that if $(x, r, p, X) \to F(t, x, r, p, X)$ is proper for a fixed $t \in [0, T]$, then so is the associated parabolic problem:

$$\phi_t + F(t, x, \phi, D\phi, D^2\phi) = 0.$$

Let us suppose that $\phi \in C^2(O)$ and that $F(x, \phi(x), D\phi(x), D^2\phi(x)) \leq 0$, $\forall x \in R^n$. Suppose that φ is also $C^2(O)$ (O an open subset of R^n) and $\widetilde{x} \in O$ is a local maximum

of $\phi - \varphi$. This implies that $D(\phi - \varphi)(\widetilde{x}) = 0$ and $D^2(\phi - \varphi)(\widetilde{x}) \leq 0$, so

$$\left\{ \begin{array}{l} D(\phi)(\widetilde{x}) = D(\varphi)(\widetilde{x}) \\ D^2(\phi)(\widetilde{x}) \leq D^2(\varphi)(\widetilde{x}) \end{array} \right. .$$

Using the property of degenerate ellipticity of F, we get

$$F(\widetilde{x}, \phi(\widetilde{x}), D\varphi(\widetilde{x}), D^2\varphi(\widetilde{x})) \leq F(\widetilde{x}, \phi(\widetilde{x}), D\phi(\widetilde{x}), D^2\phi(\widetilde{x})) \leq 0.$$

For a given proper function establishing the uniqueness of solutions requires some extra structural conditions (or assumptions) on the equation. The three basic properties of viscosity solutions are existence, uniqueness and stability. The existence of solutions holds in all cases where the comparison principle holds and the boundary conditions can be enforced in some way as shown in the following. More definition of viscosity solutions can be found in Appendix 1 and the references therein. Here the existence theorem for viscosity solutions will be based on the one introduced by Ishii and Sato

[1] A given PDE

$$E(x, u, Du, D^2u) = 0 \tag{8.12}$$

over a domain $x \in \Omega$ has a solution in the classical concept if we can find a function $u(x)$ which is continuous and differentiable over the entire domain such that x, u, Du, D^2u satisfy the above equation at every point.

in [78], assuming that the given domain Ω is a bounded in R^n with a C^1 boundary.

Let us consider the following conditions:

(**H1**) $F \in C([0,T] \times \overline{\Omega} \times R \times (R^n - \{0\}) \times S^n)$, where S^n denotes the space of $n \times n$ symmetric matrices equipped with the usual ordering.

(**H2**) There exists a constant $\gamma \in R$ such that for each $(t,x,p,X) \in [0,T] \times \overline{\Omega} \times R \times (R^n - \{0\}) \times S^n)$, the function $\phi \to F(t,x,\phi,p,X) - \gamma\phi$ is non decreasing on R.

(**H3**) For each $Q > 0$, there exists a continuous function $w_Q : [0,\infty[\to [0,\infty[$ satisfying $w_Q(0) = 0$ such that if $X, Y \in S^n$ and $\mu_1, \mu_2 \in [0,\infty[$ satisfy:

$$\begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \leq \mu_1 \begin{pmatrix} I & -I \\ -I & I \end{pmatrix} + \mu_2 \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \tag{8.14}$$

then

$$\begin{aligned} F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y) \geq -w_Q\Big(\mu_1(|x-y|^2 + \rho(p,q)^2) + \mu_2 \\ +|p-q| + |x-y|(1 - \max(|p|,|q|))\Big) \end{aligned} \tag{8.15}$$

for all $t \in [0,T], x, y \in \overline{\Omega}, \phi \in R$ with $|\phi| \leq Q$ and $p, q \in R^n \backslash \{0\}$.

(**H4**) $B \in C(R^n \times R^n) \cap C^{1,1}(R^n \times (R^n \backslash \{0\}))$.

(**H5**) For each $x \in R^n$, the function $p \to B(x,p)$ is positively homogeneous of degree one in p, i.e., $B(x,\lambda p) = \lambda B(x,p), \forall \lambda \geq 0, p \in R^n \backslash \{0\}$.

(**H6**) There exists a positive constant θ such that $< \upsilon(z), D_p B(z,p) \geq \theta$ for all $z \in \partial\Omega$ and $p \in R^n - \{0\}$. Here $\upsilon(z)$ denotes the unit outer normal vector of at $z \in \partial\Omega$.

We recall the following theorem taken from [67].

Theorem 8.3.1 *Consider the following problem:*

$$\begin{cases} \phi(0,x) = \phi_0(x), \quad \textit{for } x \in \overline{\Omega} \\ \phi_t + F(t,x,\phi,D\phi,D^2\phi) = 0 \ \textit{in }]0,T[\times\Omega \ , \\ B(x,Du) = 0 \quad \textit{in }]0,T[\times\partial\Omega \ , \end{cases} \tag{8.16}$$

Assume that conditions H1–H6 hold. Then for each $\phi_0 \in C(\overline{\Omega})$ there is a unique viscosity solution $\phi \in C([0,T[\times\overline{\Omega})$ of (8.16) satisfying $\phi(0,x) = \phi_0(x)$ for $x \in \overline{\Omega}$.

8.3.2 Existence and Uniqueness Results for *Model 1*

Checking the Properness of F, *Model 1*.

Coming back to *Model 1,* the stated problem is given in the form

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon(\phi)\Big\{\mu\nabla\cdot\left(W(\mathbf{x})\frac{\nabla\phi}{|\nabla\phi|}\right) - \Big[\lambda_1(u_0(\mathbf{x}) - c_1)^2 - \lambda_2(u_0(\mathbf{x}) - c_2)^2\Big] -$$

$$\nu\Big[\Big(\int_\Omega H_\epsilon(\phi)d\mathbf{x} - A_1\Big) - \Big(\int_\Omega (1 - H_\epsilon(\phi))d\mathbf{x} - A_2\Big)\Big]\Big\} - \alpha W\mathbf{x}|\nabla\phi| = 0, \tag{8.17}$$

with $\mathbf{x} = (x,y)$. As done in [39, 182] a rescaling can be made so that the motion is applied to all level sets by replacing $\delta_\epsilon(\phi)$ by $|\nabla\phi|$

$$\begin{cases} \phi(\mathbf{x},0)=\phi^0(\mathbf{x}) \\ \frac{\partial\phi}{\partial t}=\mu|\nabla\phi|W(\mathbf{x})\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)+\mu<\nabla W(\mathbf{x}),\nabla\phi>- \\ \qquad |\nabla\phi|T_1(u_0,c_1,c_2)-|\nabla\phi|T_2(H_\epsilon(\phi),A_1,A_2)-\alpha W(\mathbf{x})|\nabla\phi|, \\ \left.\frac{\partial\phi}{\partial\upsilon}\right|_{\partial\Omega}=0, \end{cases} \tag{8.18}$$

where $T_1(u_0,c_1,c_2)=\lambda_1(u_0(\mathbf{x})-c_1)^2-\lambda_2(u_0(\mathbf{x})-c_2)^2, T_2(H_\epsilon(\phi),A_1,A_2)=\nu\Big[\Big(\int_\Omega H_\epsilon(\phi)dxdy-A_1\Big)-\Big(\int_\Omega(1-H_\epsilon(\phi))dxdy-A_2\Big)\Big]$. The function $g(\nabla u_0)$ bounded continuous and $\nabla W(\mathbf{x})=\nabla g(\nabla u_0)$ (or $\nabla W(\mathbf{x})=\nabla(d(\mathbf{x})g(\nabla u_0))$ is well-defined (except at the points that are equidistant from at least two points of the given finite set $\mathcal{A}$, see [66]).

In this way we have a parabolic PDE of the form

$$\phi_t+F(t,x,\phi,D\phi,D^2\phi)=0,\ \text{in}\ [0,T]\times\Omega.$$

with

$$F(t,x,\phi,D\phi,D^2\phi)=-\mu|D\phi|W(\mathbf{x})D\cdot\left(\frac{D\phi}{|D\phi|}\right)-\mu<\nabla W(\mathbf{x}),D\phi>+ \\ |D\phi|T_1(u_0,c_1,c_2)+|D\phi|T_2(H(D\phi),A_1,A_2)+\alpha W(x,y)|D\phi|.$$

Before checking if these conditions are satisfied by the model we rewrite the equation in the form

$$F(t,x,r,p,X)=-\mu W(\mathbf{x})trace\Big(A(p)X\Big)-\mu<\nabla W(\mathbf{x}),p>+ \\ |p|T_1(u_0,c_1,c_2)+|p|T_2(H_\epsilon(r),A_1,A_2)+\alpha W(\mathbf{x})|p|$$

with $A(p)=I-\frac{p\otimes p}{|p|^2}$ (for details refer to Appendix 2).

In order the first condition (**P1**) to be satisfied we have to analyze the term $T_2(H_\epsilon(r),A_1,A_2)$ since the other terms does not depend explicitly on r. With the assumption that the initialization is placed inside the object, while the curve will evolve towards the boundaries the area of the region inside the curve grows while outside decreases. In this way the integrals $\int_\Omega H_\epsilon(r)dxdy-A_1\geq 0$ and $-\int_\Omega(1-H_\epsilon(r))dxdy-A_2\Big)\geq 0$, and $\nu\Big[\Big(\int_\Omega H_\epsilon(r)dxdy-A_1\Big)-\Big(\int_\Omega(1-H_\epsilon(r))dxdy-A_2\Big)\Big]\geq 0$. For more if $s\geq r$ taking in consideration that $\int_\Omega H_\epsilon(\phi)dxdy$ represents the area of a given curve we have $\int_\Omega H_\epsilon(s)dxdy-A_1\geq\int_\Omega H_\epsilon(r)dxdy-A_1$ and $-\Big(\int_\Omega(1-H_\epsilon(s))dxdy-A_2\Big)\geq -\Big(\int_\Omega(1-H_\epsilon(r))dxdy-A_2\Big)$. This means $T_2(H_\epsilon(s),A_1,A_2)\geq T_2(H_\epsilon(r),A_1,A_2)\Rightarrow F(x,s,p,X)\geq F(x,r,p,X)$.

Let us analyze the degenerate ellipticity (**P2**). We have then to compare $F(x,y,p,X)$ and $F(x,y,p,Y)$ when $Y\geq X$ symmetric matrices[2]. Since the other terms do not in-

[2] "$\geq$" stands for the usual partial ordering on symmetric matrices stating that $Y\geq X$ if $Y-X\geq 0$ i.e., $Y-X$ is positive semi-definite

volve the second derivative, it has to be compared

$$-\mu W(\mathbf{x})trace\Big(A(p)X\Big) \text{ and } \quad -\mu W(\mathbf{x})trace\Big(A(p)Y\Big)$$

when $Y \geq X$. With some simple algebra it can be shown that (see Appendix 3)

$$-\mu W(\mathbf{x})trace(AY) \leq -\mu W(\mathbf{x})trace(AX).$$

In this way it was proved that for $p \neq 0$, F is degenerate elliptic. We then suppose for the following F is proper and continuous.

Existence and Uniqueness of the Solution *Model 1*: Proof Based on Ishii and Sato's Work [78]

Before starting we remind that the function $\int_\Omega H_\epsilon(\phi)dxdy$ and $\int_\Omega (1 - H_\epsilon(\phi))dxdy$ are continuous bounded functions. This mean that $T_2(H_\epsilon(\phi), A_1, A_2)$ is Lipschitz.

Going back to our problem *Model 1*, we can apply the viscosity theorem. The considered *Model 1* problem functional F is defined by

$$F(t, x, \phi, p, X) = -\mu W(x, y)trace\Big(A(p)X\Big) - \mu < \nabla W(x, y), p > + \\ |p|T_1(u_0, c_1, c_2) + |p|T_2(H(\phi), A_1, A_2) + \alpha W(x, y)|p|$$

Theorem 8.3.2 *Consider the following problem:*

$$\begin{cases} \phi(0, x) = \phi_0(x), \quad for\ x \in \overline{\Omega} \\ \phi_t + F(t, x, \phi, p, X) = 0 \ in\ ]0, T[\times\Omega\ , \\ B(x, p) = 0 \quad\ \ in\ ]0, T[\times\partial\Omega\ , \end{cases} \tag{8.19}$$

For each $\phi_0 \in C(\overline{\Omega})$ the conditions H1–H6 hold, which garanties a unique viscosity solution $\phi \in C([0, T[\times\overline{\Omega})$ of (8.19) satisfying $\phi(0, x) = \phi_0(x)$ for $x \in \overline{\Omega}$.

Proof The first point (**H1**) is satisfied because F presents a singularity for $p = 0$ but is continuous otherwise.

The second condition (**H2**): The term $|p|T_1(u_0, c_1, c_2)$ does not depend explicitly on ϕ, for this reason γ has to be chosen such that $|p|T_2(H(\phi), A_1, A_2)$ is non decreasing[3]. As the level set tends to the real boundaries of the object the term $T_2(H(\phi), A_1, A_2)$ is a positive evaluated constant so we can choose a negative constant γ such that $T_2(H(\phi), A_1, A_2) - \gamma\phi$ is non decreasing. In this way the second condition is satisfies (since if $f(b) \geq f(a)$ for $b > a$, then $f(b) + c \geq f(a)$ for c positive).

The third condition (**H3**): Using the inequality

$$trace(A(p)X) + trace(A(q)Y) \leq \mu_1(2\xi|x - y|^2 + 8\theta\rho(p, q)^2) + 2\theta\mu_2$$

[3]A function is non decreasing if $f(b) \geq f(a)$ for $b > a$.

provided in Appendix 4 we can evaluate the expression

$$\begin{aligned} F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y) = & -\mu(trace(A(p)X) + trace(A(q)Y)) - \\ & \mu(< \nabla W(x), p > - < \nabla W(y), q >) + \\ & (|p| - |q|)T_1(u_0, c_1, c_2) + \\ & (|p|T_2(H(p), A_1, A_2) - |q|T_2(H(q), A_1, A_2) \\ & + \alpha W(x,y)(|p| - |q|) \end{aligned}$$

Using the same arguments as in [69], we have

$$< \nabla W(x), p > - < \nabla W(y), q > \leq C_1|x-y| \max(|p|,|q|) + C_2|p-q|$$

and

$$(|p| - |q|)T_1(u_0, c_1, c_2) + \alpha W(x,y)(|p| - |q|) \leq (\theta_1 + \theta_2)|p-q|$$

$$\begin{aligned} |p|T_2(H(p), A_1, A_2) - |q|T_2(H(q), A_1, A_2) & \leq |p| - |q| \max(T_2(H(p), A_1, A_2), T_2(H(q), A_1, A_2)) \\ & \leq \theta_3|p-q| \end{aligned}$$

from which we deduce

$$\begin{aligned} -(F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y)) \leq & \mu_1(2\xi|x-y|^2 + 8\theta\rho(p,q)^2) + 2\theta\mu_2 + \\ & C_1|x-y| \max(|p|,|q|) + C_2|p-q| - \\ & (\theta_1 + \theta_2)|p-q| - \theta_3|p-q| \end{aligned} \tag{8.20}$$

and

$$\begin{aligned} -(F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y)) \leq & \max(2\xi, 8\theta, C_1, C_2 - \theta_1 - \theta_2 - \theta_3) \Big[\\ & \mu_1(|x-y|^2 + \rho(p,q)^2) + \mu_2 + \\ & (1 + |x-y|) \max(|p|,|q|) + |p-q| \Big] \end{aligned}$$

We just have to take $w_Q(l) = \max(2\xi, 8\theta, C_1, C_2 - \theta_1 - \theta_2 - \theta_3)l$. $w_Q(0) = 0$ and w_Q is nondecreasing on $[0, \infty[$.

Similar to [66], the fourth point (**H4**) is fulfilled with assumptions on υ (8.21). Then, it is easy to check that B is positively homogeneous of degree one (**H5**). For the last point (**H6**), one can easily see that

$$B(z,p) = < \upsilon(z), p > \tag{8.21}$$

and

$$< \upsilon(z), D_p B(z,p) > = |\upsilon(z)|^2 = 1 \tag{8.22}$$

We take $\theta = 1$ and the last assumption is fulfilled. □

8.3.3 Existence and Uniqueness Results for *Model 2*

Checking the Properness of F, *Model 2*

In the second case the c_2 term has been updated in each iteration, which means it is a term of the form $c_2(\phi)$. The stated problem in this case is:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\Big\{\mu\nabla\cdot\left(W(\mathbf{x})\frac{\nabla\phi}{|\nabla\phi|}\right) - \Big[\lambda_1(u_0(\mathbf{x})-c_1)^2 - \lambda_2(u_0(\mathbf{x})-c_2(\phi))^2\Big] -$$

$$\nu\Big[\Big(\int_\Omega H_\epsilon(\phi)dxdy - A_1\Big) - \Big(\int_\Omega (1-H_\epsilon(\phi))dxdy - A_2\Big)\Big]\Big\} - \alpha W\mathbf{x}|\nabla\phi| = 0, \quad (8.23)$$

with $\mathbf{x} = (x, y)$. Replacing $\delta_\epsilon(\phi)$ by $|\nabla\phi|$ in a partial scaling excluding the term $T_1(u_0, c_1, c_2(\phi))$, for analysis purpose, we have

$$\begin{cases} \phi(\mathbf{x}, 0) = \phi^0(\mathbf{x}) \\ \frac{\partial\phi}{\partial t} = \mu|\nabla\phi|W(\mathbf{x})\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + < \nabla W(\mathbf{x}), \nabla\phi > - \\ \qquad \delta_\epsilon(\phi)T_1(u_0, c_1, c_2(\phi)) - |\nabla\phi|T_2(H_\epsilon(\phi), A_1, A_2) - \alpha W(\mathbf{x})|\nabla\phi|, \\ \frac{\partial\phi}{\partial\vec{n}}\Big|_{\partial\Omega} = 0, \end{cases} \quad (8.24)$$

with $T_1(u_0, c_1, c_2) = \lambda_1(u_0(\mathbf{x})-c_1)^2 - \lambda_2(u_0(\mathbf{x})-c_2(\phi))^2, T_2(H_\epsilon(\phi), A_1, A_2) = \nu\Big[\Big(\int_\Omega H_\epsilon(\phi)dxdy - A_1\Big) - \Big(\int_\Omega(1-H_\epsilon(\phi))dxdy - A_2\Big)\Big]$.

As before, we have

$$\phi_t + F(t, \mathbf{x}, \phi, D\phi, D^2\phi) = 0, \text{ in } [0, T]\times\Omega$$

where F is defined as $F : R^n \times R \times R^n \times S(n) \to R$ with the specific form

$$F(t, \mathbf{x}, \phi, p, X) = -\mu W(x, y)trace\Big(A(p)X\Big) - \mu\langle\nabla W(x, y), p\rangle + \\ \delta_\epsilon(\phi)T_1(u_0, c_1, c_2(\phi)) + |p|T_2(H(\phi), A_1, A_2) + \alpha W(x, y)|p| \quad (8.25)$$

and $S(n)$ denotes the set of symmetric matrices.

First we verify if F is proper by checking if the condition (**P1**) and (**P2**) are satisfied. For the first condition we have to analyze the term $\delta_\epsilon(\phi)T_1(u_0, c_1, c_2(r)) + |\nabla\phi|T_2(H_\epsilon(r), A_1, A_2)$, because for the rest we can use the same arguments as in Subsection §8.3.2. The difference now is that the expression $T_1(u_0, c_1, c_2(r)) = \lambda_1(u_0(\mathbf{x}) - c_1)^2 - \lambda_2(u_0(\mathbf{x}) - c_2(r)^2)$, updated each iteration the term c_2 depends on $H_\epsilon(r)$. To carry with the check of the condition let's consider a simple case of one object and its background that $c_1 < c_2$. This assumption does not constrict the class of the images we deal with because in the case of brighter background in relation to the object the model can be adapted by processing the $255 - u_0$ instead of u_0. While the curve evolves in each iteration k the term c_2^k decays by getting closer to the the real background intensity c_2^* , where c_2^k denotes c_2 in each time step of computation and c_2^* the correct answer for mean intensity of background. On the other hand this means that if we consider $s \geq r$, $c_2(s) \leq c_2(r)$, this is due to the fact that the curve representing s will be less corrupted

with elements from the object with lower intensity. In this context

$$
\begin{array}{c}
0 \leq c_2(s) \leq c_2(r) \Rightarrow \\
0 \leq c_2(s) - u_0(\mathbf{x}) \leq c_2(r) - u_0(\mathbf{x}) \text{ for the foreground} \\
u_0(\mathbf{x}) - c_2(s) \geq u_0(\mathbf{x}) - c_2(r) \ \geq 0 \text{ for the background} \\
\\
(u_0(\mathbf{x}) - c_2(s))^2 \leq (u_0(\mathbf{x}) - c_2(r))^2 \text{ for the foreground} \\
(u_0(\mathbf{x}) - c_2(s)^2 \geq ((u_0(\mathbf{x}) - c_2(r))^2 \text{ for the background} \\
\\
-(u_0(\mathbf{x}) - c_2(s))^2 \geq -(u_0(\mathbf{x}) - c_2(r))^2 \text{ for the foreground} \\
-(u_0(\mathbf{x}) - c_2(s))^2 \leq -((u_0(\mathbf{x}) - c_2(r))^2 \text{ for the background}
\end{array}
$$

At this point we have to remind that the term $T_1(u_0, c_1, c_2(r))$ appears multiplied by $\delta_\epsilon(r)$. This means that the terms $-(u_0(\mathbf{x}) - c_2(s))^2$ and $-(u_0(\mathbf{x}) - c_2(r))^2$ will be zero everywhere except in a small $\epsilon-$neigborhood of s and r. Moreover, we consider that r and s are evolving until reaching the boundaries and are $\epsilon-$distance from the target object. This means that $-(u_0(\mathbf{x}) - c_2(s))^2 \geq -(u_0(\mathbf{x}) - c_2(r))^2$. Regarding the second term $T_2(H_\epsilon(r), A_1, A_2)$ we conclude same as before that

$$T_2(H_\epsilon(s), A_1, A_2) \geq T_2(H_\epsilon(r), A_1, A_2) \Rightarrow F(x,s,p,X) \geq F(x,r,p,X).$$

The analyzes for degenerate ellipticity condition (**P2**) is same like done in Subsection §8.3.2.

We can prove the existence and uniqueness of the solution *Model 2* in an $\epsilon-$distance neigborhood of the final aimed contour, in which the function F is proper.

Existence and Uniqueness of the Solution *Model 2*: Proof Based on Ishii and Sato's Work [78]

Going back to our problem, we can apply the viscosity theorem. The considered Model 1 problem functional F is defined by

$$
\begin{array}{c}
F(t,x,\phi,p,X) = -\mu W(x,y) trace\Big(A(p)X\Big) - \mu < \nabla W(x,y), p > + \\
\delta_\epsilon(\phi) T_1(u_0, c_1, , c_2(\phi)) + |p| T_2(H(\phi), A_1, A_2) + \alpha W(x,y)|p|
\end{array}
$$

Theorem 8.3.3 *Consider the following problem:*

$$
\left\{
\begin{array}{l}
\phi(0,x) = \phi_0(x) \ \ \textit{for } x \in \overline{\Omega} \\
\phi_t + F(t,x,\phi,p,X) = 0 \ \textit{in }]0,T[\times\Omega \ , \\
B(x,p) = 0 \quad \textit{in }]0,T[\times\partial\Omega \ ,
\end{array}
\right. \tag{8.26}
$$

For each $\phi_0 \in C(\overline{\Omega})$ the conditions H1–H6 hold, which guaranties a unique viscosity solution $\phi \in C([0,T[\times\overline{\Omega})$ of (8.26) satisfying $\phi(0,x) = \phi_0(x)$ for $x \in \overline{\Omega}$.

Proof It suffices to verify conditions (**H1** − **H6**) in order to apply Theorem 8.3.1. The first point (**H1**) is satisfied because F presents a singularity for $p = 0$ but is continuous otherwise.

For the second condition (**H2**), γ has to be chosen such that $|p|T_2(H(\phi), A_1, A_2) + \delta_\epsilon(\phi)T_1(u_0, c_1, c_2(\phi))$ is non decreasing. As the level set tends to the real boundaries of the given the term $T_2(H(\phi), A_1, A_2)$ is an positive evaluated constant. The term

$T_1(u_0, c_1, c_2(\phi))$ is a bounded value, so we can choose a negative constant γ such that $\delta_\epsilon(\phi)T_1(u_0, c_1, c_2(\phi)) - \gamma\phi$ non decreasing, so satisfies the second condition.

In the same way like Subsection §8.3.2 for the third condition (**H3**) we have

$$trace(A(p)X) + trace(A(q)Y) \leq \mu_1(2\xi|x-y|^2 + 8\theta\rho(p,q)^2) + 2\theta\mu,$$

$$< \nabla W(x), p > - < \nabla W(y), q > \leq C_1|x-y|\max(|p|,|q|) + C_2|p-q|.$$

Evaluating the expression

$$\begin{aligned} F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y) = &-\mu(trace(A(p)X) + trace(A(q)Y)) - \\ &(< \nabla W(x), p > - < \nabla W(y), q >) + \\ &\delta_\epsilon(\phi)(T_1(u_0, c_1, c_2(\phi)) - T_1(u_0, c_1, c_2(\phi))) + \\ &(|p|T_2(H(p), A_1, A_2) - |q|T_2(H(q), A_1, A_2) \\ &+ \alpha W(x,y)(|p| - |q|) \end{aligned}$$

Since

$$\delta_\epsilon(\phi)\cdot 0 + \alpha W(x,y)(|p| - |q|) \leq \theta_1|p-q|$$

$$\begin{aligned} |p|T_2(H(p), A_1, A_2) - |q|T_2(H(q), A_1, A_2) &\leq |p| - |q|\max(T_2(H(p), A_1, A_2), T_2(H(q), A_1, A_2)) \\ &\leq \theta_2|p-q| \end{aligned}$$

we deduce

$$\begin{aligned} -(F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y)) \leq &\mu_1(2\xi|x-y|^2 + 8\theta\rho(p,q)^2) + 2\theta\mu_2 + \\ &C_1|x-y|\max(|p|,|q|) + C_2|p-q| - \\ &(\theta_1+\theta_2)|p-q| \end{aligned} \tag{8.27}$$

and

$$\begin{aligned} -(F(t,x,\phi,p,X) - F(t,y,\phi,q,-Y)) \leq \max(2\xi, 8\theta, C_1, C_2 - \theta_1 - \theta_2)\Big[\\ \mu_1(|x-y|^2 + \rho(p,q)^2) + \mu_2 + \\ (1 + |x-y|)\max(|p|,|q|) + |p-q|\Big] \end{aligned}$$

We just have to take $w_R(l) = \max 2\xi, 8\theta, C_1, C_2 - \theta_1 - \theta_2)l.wR(0) = 0$ and $w_R = 0$ is nondecreasing on$[0, \infty[$.

The fourth point is fulfilled with assumptions on υ (8.28). Then, it is easy to check that B is positively homogeneous of degree one. For the last point, one can easily see that

$$B(z,p) = < \upsilon(z), p > \tag{8.28}$$

and

$$< \upsilon(z), D_pB(z,p) > = |\upsilon(z)| = 1 \tag{8.29}$$

We take $\theta = 1$ and the last assumption is fulfilled. □

8.4 Numerical Solution: An AOS Algorithm

Recalling equation (8.11)

$$\begin{cases} \phi(x,y,0) = \phi^0(x,y) \\ \frac{\partial \phi}{\partial t} = \mu\delta_\epsilon(\phi)\nabla\cdot\left(W\frac{\nabla\phi}{|\nabla\phi|_\beta}\right)+ \\ \qquad \delta_\epsilon(\phi)\Big\{-\Big[\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big]- \\ \qquad \nu\Big[\Big(\int_\Omega H dxdy - A_1\Big)-\Big(\int_\Omega(1-H)dxdy - A_2\Big)\Big]\Big\} - \alpha W(x,y)|\nabla\phi|, \\ \frac{\partial\phi}{\partial\vec{n}}\Big|_{\partial\Omega} = 0, \end{cases} \tag{8.30}$$

we can implement a fast and a low computational cost method such as AOS.

Denoting $E = \frac{W}{|\nabla\phi|_\beta}$, and

$$\begin{aligned} f =& \delta_\epsilon(\phi)\Big\{-\Big[\lambda_1(u_0(x,y)-c_1)^2-\lambda_2(u_0(x,y)-c_2)^2\Big]- \\ & \nu\Big[\Big(\int_\Omega H dxdy - A_1\Big)-\Big(\int_\Omega(1-H)dxdy - A_2\Big)\Big]\Big\}- \\ & \alpha W(x,y)|\nabla\phi|, \end{aligned} \tag{8.31}$$

equations (8.30) can be written in the compact form:

$$\begin{cases} \dfrac{\partial\phi}{\partial t} = \mu\delta_\epsilon(\phi)\nabla\cdot(E\nabla\phi) + f = \\ \qquad \mu\delta_\epsilon(\phi)(\partial_x(E\partial_x\phi)+\partial_y(E\partial_y\phi)) + f. \end{cases} \tag{8.32}$$

Discretizing in the spatial step, the equation (8.30) one can rewrite it in the matrix-vector form which, by employing the AOS scheme, can be split additively as shown below:

$$\phi^{n+1} = \frac{1}{2}\sum_{l=1}^{2}(\ I - 2\Delta t A_l(\phi^n))^{-1}\hat{\phi}^n \tag{8.33}$$

where the tridiagonal matrices A_l for $l = 1, 2$ are derived in the same way as in Chapter 5.

The following iteration algorithm has been designed for the model:

8.5 Experimental Results

In order to illustrate the performance and the accuracy of the proposed method, experiments were carried out on synthetic, CT and MRI 2-D images. We like to emphasize that for the new method the distance function was not considered in none of the results shown below, in other words $d(x,y) = 1$.

We will present three set of experiments. In the first one we demonstrate that our new segmentation method works for segmenting hard cases, such as the objects shown in Fig. 5.2, and compare the results with a *one level set model* of the same type, such as the Badshah-Chen model. The second set will compared the new model with the Nguyen-Cai-Zhang-Zheng model [8] which is known for its robustness, accurate

Algorithm 4 AOS Method Algorithm for Solving the One Level Selective Method: $\phi^k \leftarrow OLSS(\phi^{(0)}, \mathcal{A}, \mu, \nu, \beta, \alpha, \epsilon, maxit, tol)$.

Calculate the edge based function and area of the polygon (distance function optional);
$n = 1$, Compute f from equation (8.31), $\phi^{(1)} = \phi^{(0)}$;
for $iter = 1 : maxit$ **do**
Compute $\phi^{(n)}$ using (8.33):

$$\phi_i^{(n+1)} \leftarrow \frac{1}{2} \sum_{l=1}^{2} (\ I - 2\Delta t A_l(\phi^n))^{-1} \hat{\phi}^n$$

If $\|\phi^{(n+1)} - \phi^{(n)}\| < tol$ or $iter > maxit$, set $\phi^{(k)} \leftarrow \phi^{(n-1)}$ Break;
else $\phi^{(k)} \leftarrow \phi^{(n-1)}$
update f from equation (8.31)

end for

boundary with a small amount of user interaction. It will be noticed that our model performs better then Nguyen-Cai-Zhang-Zheng model in some cases where the objects have same intensity and the boundaries are vague or object with same intensity. The third set will present a comparison with Rada-Chen [129], Chapter 5, which has shown for good performance in selecting the right object in cases of low intensity difference or with objects which are close to each other. For most of this images the old models of Badshah-Chen and Gout-Guyader were considered as challenging problems.

In the following experiments the parameters Δt, λ_1, λ_2, α, h (step size), ϵ and β, have been fixed at $\Delta t = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\alpha = -0.01$, $h = 1$, $\epsilon = 1$, and $\beta = 10^{-6}$, respectively. Differently sized images $n \times n = 128 \times 128$, 256×256 have been tested and show the same satisfactory results.

Through the experiments it was observed that the parameters μ, ν can be in a range between $\mu = 100$ to $n^2/10$ and $\nu = 0.1$ to 1 which gave similar results. In hard cases with small intensity difference between the objects the parameter k can be increased in a value greater than 100, giving a better performance in this way.

The first initial level set has been constricted as a sign distant function of the given polygon constructed with the markers as more proper in our experiments. The experimental results show that if the markers are outside the object but not inside nearby objects the method works, in contrast with the case where the markers are placed near the boundaries or inside other objects in which the method fails by capturing the nearby objects.

8.5.1 Test Set 1 — robustness and accuracy of the new model, and comparison with Badsah-Chan model

In the first test set, we demonstrate the ability of recognizing objects which have a small intensity difference. Figure 8.1 shows a rectangle and triangle with a non empty intersection and with small intensity difference of those two objects. The Figure 8.1 (c) and (d) presents the satisfactory experimental results using the new model algorithm for capturing both of these objects separately, while the old Badshah-Chan model would easily fail in this case, as shown in Figure 8.1(b), where all black area was segmented. The real life image or the medical images are a real challenge for segmentation in general and especially for selective segmentation, due to poor quality and noise. In

Figure 8.2(c) and (d) we show that the model works satisfactorily with this kind of images. All these images have been shown as hard cases in the Rada-Chen [129] paper and segmented correctly by a dual level set, results which are the same with the one we already showed in Figure 8.1 (c), (d), Figure 8.2(c) and (d).

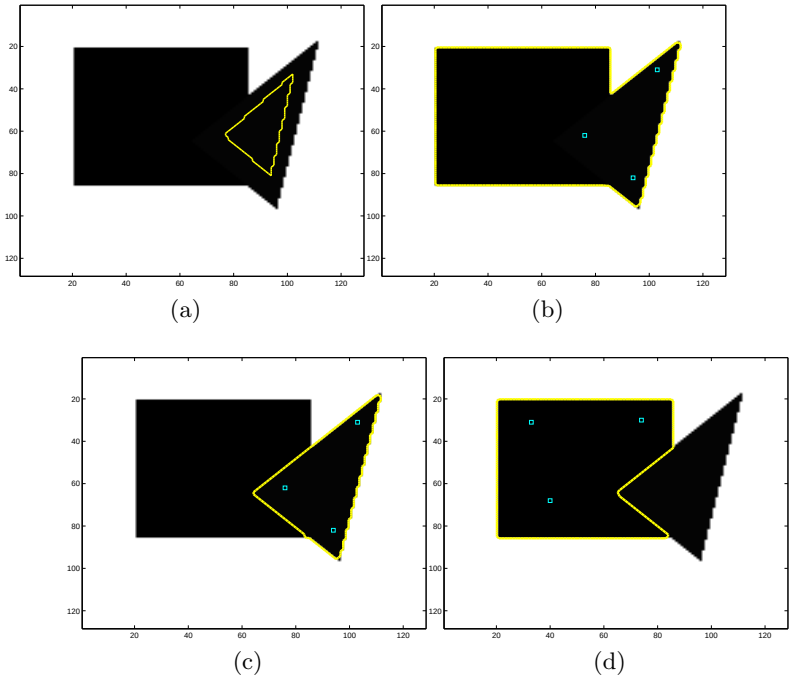


Figure 8.1: Test Set 1 – Comparison with Badshah-Chen model [12]: (a) First level set; (b) Unsuccessful result by [12] model for the case of two object with small intensity difference; (c-d) Successful result by new model for the case of two objects with small intensity difference

Test Set 2 — comparison with the previous Nguyen-Cai-Zhang-Zheng model [8]

We have to acknowledge that Nguyen-Cai-Zhang-Zheng model [8] gives same satisfactory results for the test images in Fig. 8.1 and 8.2(c) and for almost all the images we present in this paper as well. Fig. 8.3 shows the successful segmentation of the kidney by Nguyen-Cai-Zhang-Zheng model in a CT image. This test set will show only a few examples where our model performs better then Nguyen-Cai-Zhang-Zheng model [8]. For briefness we will not show the results that both models give satisfactory results. Although the performance of Nguyen-Cai-Zhang-Zheng model [8] is much better then other methods, its results still contain some artifacts due to similar appearance. The method cannot handle transparent or semi-transparent boundaries. Fig. 8.4 gives a failure example, where our method is able to segment the target cell in a clean way while the Nguyen-Cai-Zhang-Zheng model [8] fails. Fig. 8.5 shows as well an unsuccessful result with objects which have same intensity while our model succeeds as shown in Fig. 8.6(a).

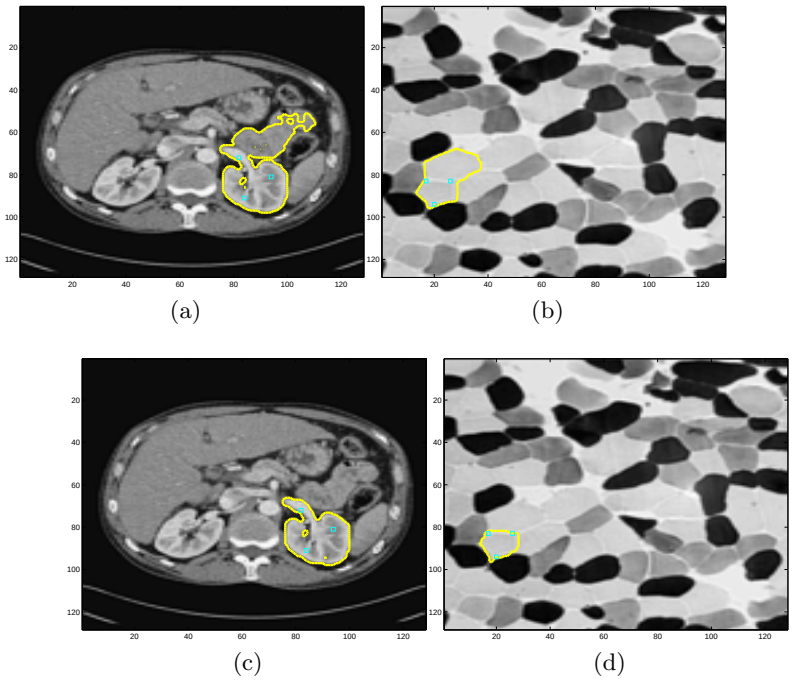
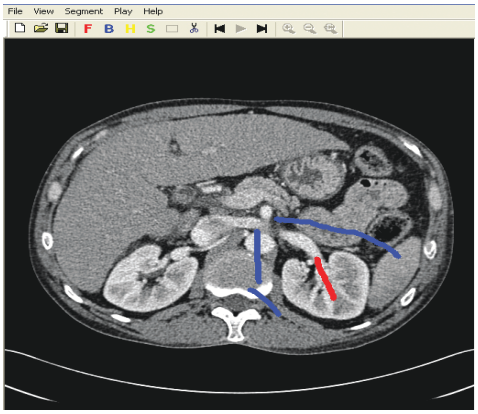


Figure 8.2: Test Set 1 – Comparison with the Badshah-Chen model [12]: (a) Unsuccessful result by [12] model for the case of CT image where the organs are nearby, with small intensity difference and in presence of noise; (b) Unsuccessful result by [12] model for the case of a biological image where the cells are of the same intensity or with small intensity difference; (c) Successful results with the new model in case of CT image where the organs are nearby, with small intensity difference and in presence of noise; (d) Successful result by new model for the case of a biological image where the cells are of the same intensity or with small intensity difference;

8.5.2 Test Set 3 — comparison with the previous Rada-Chen model

We continue our experiments by giving more examples and compare our model with the dual level set selective method [129]. Here we show 12 more different images, which can be found as successful as selected by the dual level set selective method [129]. All of the testing examples, as shown in Fig. 8.6, give satisfactory segmentation which is almost the same result as with the dual level set selective method, which for the sake of brevity we do not show. The first five images in Fig. 8.6, (a), (b), (c), (d), (e) respectively, are artificial images and from the results we conclude that the model works satisfactorily for cases where the features are nearby and with different shapes, while the last images, Figure 8.6 (f), (g), (h), (i), (j), (k), (l), show an accurate segmentation of biological and medical images, which are considered as much harder cases due to the low-quality data. Since two level set need to be updated in the case of the dual level sets, the model suffers from slow convergence. Table 8.1 shows this fact clearly. In this table we compare CPU times of the new model and the old dual level set model [129], and find out that the new model is at least two or three times faster and results obtained in all the experiments without exceptions.



(a) Two types of strokes has been labeling some foreground and background pixels

(b) Successful segmentation of the kidney by Nguyen-Cai-Zhang-Zheng model [8]

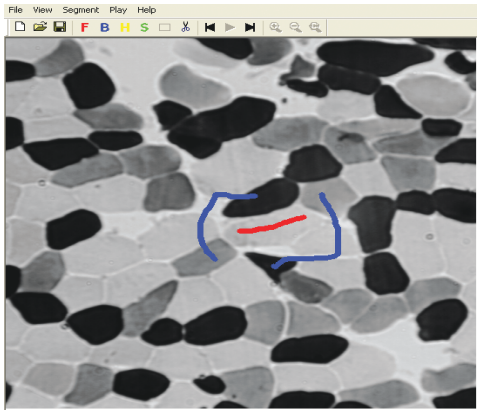
Figure 8.3: Test Set 2 – Comparison with Nguyen-Cai-Zhang-Zheng model [8]. Successful result by Nguyen-Cai-Zhang-Zheng model for the case of kidney segmentation in a CT image.

Table 8.1: Required CPU time for successful selective segmentation for some of the tested images.

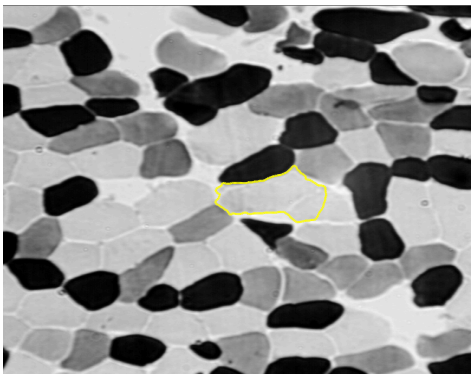
Figure	DLSS method	New method
	CPU time	CPU time
Fig.8.1(c) (256×256)	68.5938	18.7188
Fig.8.1(d) (256×256)	49.1250	18.9688
Fig.8.6(e) (128×128)	65.32	36.703
Fig.8.6(f) (128×128)	69.120	21.890
Fig.8.6(l) (128×128)	32.601	11.890

8.6 Conclusions

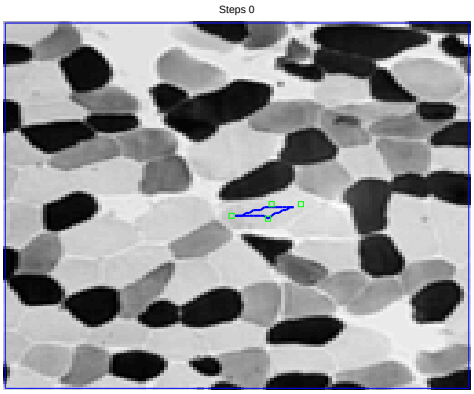
In this paper we presented a new variational selective segmentation model with one level set function which performs reliable segmentation, improving on two related models proposed recently. Existence and uniqueness of the model has been discussed on the mathematical sense of the geometric PDE model based on the viscosity theory. The new model delivers similar results for easy problems to old models, such as the Badshah-Chen model [12] or Nguyen-Cai-Zhang-Zheng model [8] and improves on this model in hard cases with objects which are nearby or have a small intensity difference, shown in different numerical experiments. The model has equally reliable results for harder problems with which the Rada-Chen [129] model could also successfully deal and at the same time improving the speed by at least two times in comparison with the model [129].



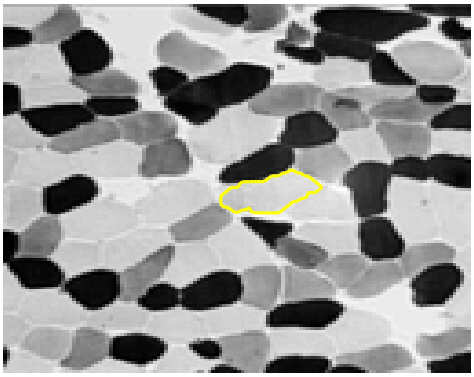
(a) Two types of strokes has been labeling some foreground and background pixels



(b) Unsuccessful result of the cell by Nguyen-Cai-Zhang-Zheng model [8]



(c) In green are 4 given markers and in blue the first level set



(d) Successful segmentation of the cell with the new model

Figure 8.4: Test Set 2 – Comparison with Nguyen-Cai-Zhang-Zheng model [8]. Successful result by new model for the case of two cells with same intensity and semi-transparent boundaries.

Appendix 1: Some Definition of Viscosity Solutions

Consider the following problem:

$$\begin{cases} \phi(0,x) = \phi_0(x), \quad \text{for } x \in \overline{\Omega} \\ \phi_t + F(t,x,\phi,D\phi,D^2\phi) = 0 \text{ in }]0,T[\times\Omega \ , \\ B(x,Du) = 0 \quad \text{ in }]0,T[\times\partial\Omega \ , \end{cases} \tag{8.34}$$

in the following we give some definition from viscosity theory.

Definition 8.6.1 *$\phi \in C(O)$ is a viscosity solution to $F = 0$ if and only if:*
$\forall \varphi \in C^2(O)$, if x_0 is a local maximum of $\phi - \varphi$, we have the relation

$$F(x_0, \phi(x_0), D\varphi(x_0), D^2\varphi(x_0)) \leq 0$$

and $\forall \varphi \in C^2(O)$, if x_0 is a local minimum of $\phi - \varphi$, we have the relation

$$F(x_0, \phi(x_0), D\varphi(x_0), D^2\varphi(x_0)) \geq 0$$

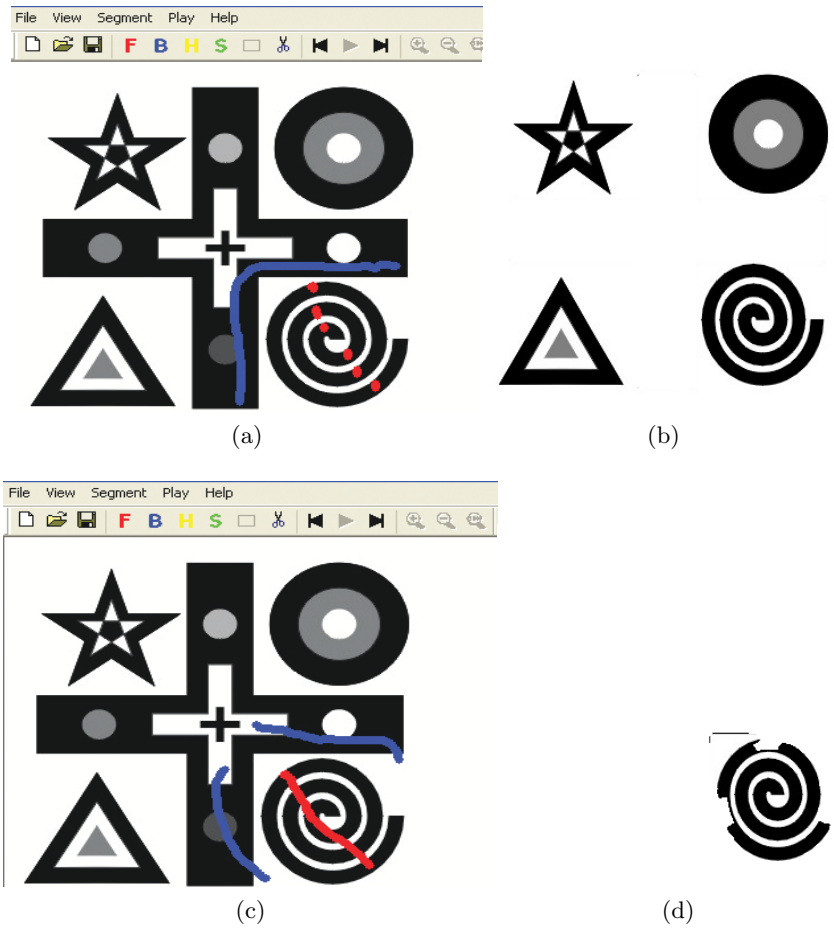


Figure 8.5: Test Set 2 – Unsuccessful result by Nguyen-Cai-Zhang-Zheng model [8] for the spiral object. The first column shows the given strokes for the foreground (in red) and background (in blue), while the second column shows the segmentation result with the Nguyen-Cai-Zhang-Zheng model [8]. Our new method shows successful result for this case as shown in Fig. 8.6(a).

If ϕ only satisfies the first (second) inequality, then ϕ is said to be a viscosity subsolution (viscosity supersolution).

Crandall et al. [48] and Barles [17] give another definition based on the notions of superjet and subjet. For x near $\widetilde{x}$, considering

$$\phi(x) - \varphi(x) \leq \phi(\widetilde{x}) - \varphi(\widetilde{x}) \;\Rightarrow \phi(x) \leq \phi(\widetilde{x}) - \varphi(\widetilde{x}) + \varphi(x)$$

and Taylor expansion gives (φ being C^2)

$$\phi(x) \leq \phi(\widetilde{x}) + < p, x - \widetilde{x} > + \frac{1}{2} < X(x - \widetilde{x}), x - \widetilde{x} > + o(|x - \widetilde{x}|^2), \; x \to \widetilde{x}, \qquad (8.35)$$

where $p = D\varphi(\widetilde{x})$ and $X = D^2\varphi(\widetilde{x})$. We say that if $\phi : O \to R$, $\widetilde{x} \in O$ and (8.35) is satisfied as $O \ni x \to \widetilde{x}$, $(p, X) \in J_o^{2,+}\phi(\widetilde{x})$, $J_O^{2,+}\phi(\widetilde{x})$ being the second order superjet of ϕ at $\widetilde{x}$.

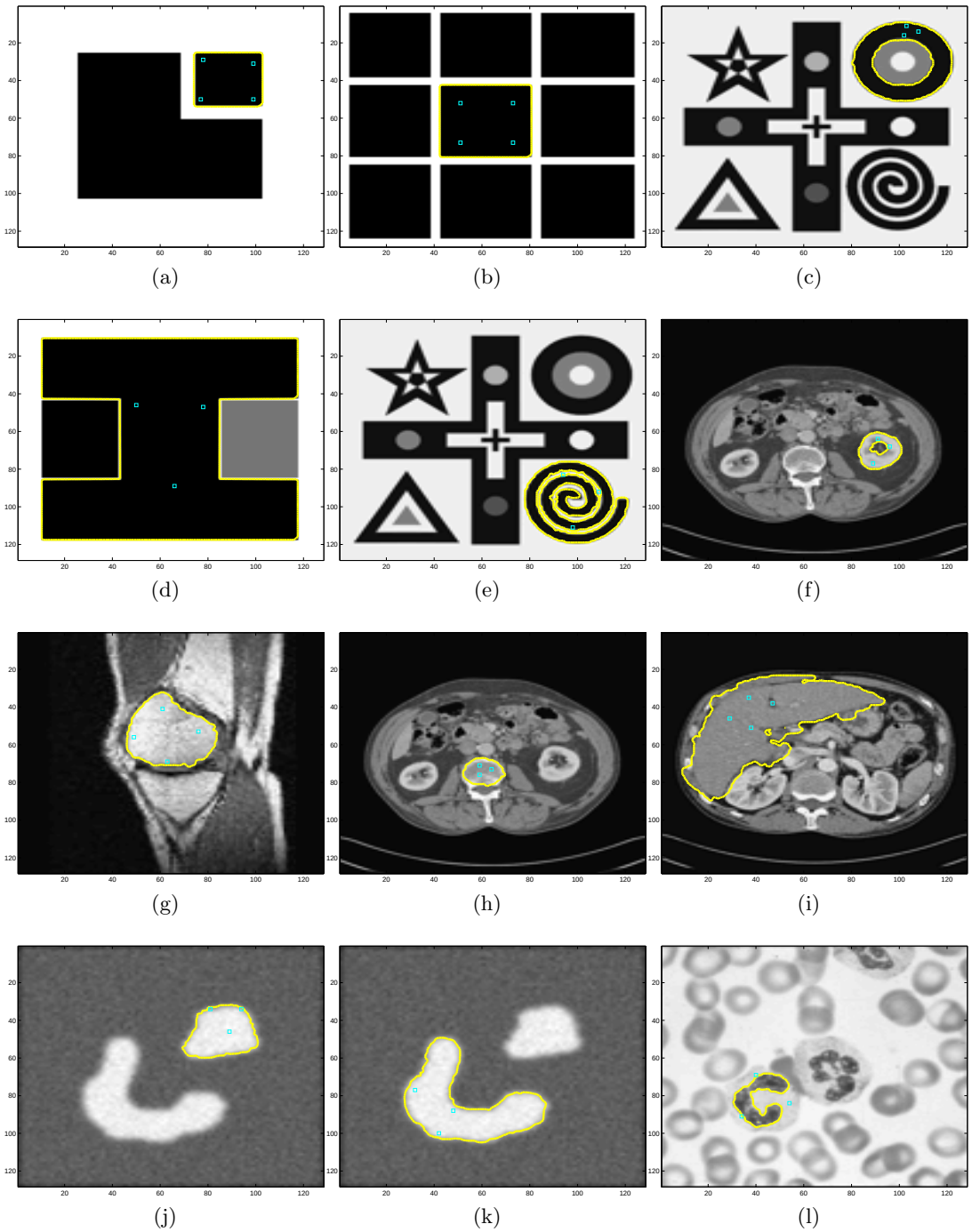


Figure 8.6: Test Set 3 – Comparative results with Rada-Chen [129] model (a-l)Successful segmentation of different images.

Definition 8.6.2 *A viscosity supersolution to $F = 0$ on O is a function $\phi \in USC(O)$*[4] *such that:*

$F(x, \phi(x), p, X) \leq 0 \ \forall x \in O, \ (p, X) \in J_O^{2+}\phi(x)$.

A viscosity supersolution to $F = 0$ on O is a function $\phi \in LSC(O)$ [5] *such that:*

$F(x, \phi(x), p, X) \geq 0 \ \forall x \in O, \ (p, X) \in J_O^{2-}\phi(x)$.

We say that ϕ is a viscosity solution if it is both a subsolution and a supersolution.

Crandall et al.[48] extend these definitions for parabolic problems

$$\phi_t + F(t, x, \phi, D\phi, D^2\phi) = 0.$$

Same with [48, 66] we assume that $D\phi$ and $D^2\phi$ stand for $D_x\phi(t, x)$ and $D_x^2\phi(t, x)$ and instead of working on O, we work on $O_T =]0, T[\times O$.

In Crandall et al.[48] $J_O^{2+}\phi(x)$ is defined as follows: $(a, p, X) \in R \times R^n \times S(n)$ lies in $J_O^{2+}\phi(x)$ if $(s, z) \in O^T$, and

$$\phi(t, x) \leq \phi(s, z) + a(t - s) + < p, t - s > + \frac{1}{2} < X(t - s), t - s > + o((t - s) + |x - z|^2),$$

as $O \ni (t, x) \to (s, z)$.

Definition 8.6.3 *A viscosity supersolution to the parabolic equation on O_T is a function $\phi \in USC(O_T)$ such that:*

$a + F(t, x, \phi(t, x), p, X) \leq 0 \ \forall (t, x) \in O_T, \ (a, p, X) \in J_O^{2+}\phi(t, x)$.

A viscosity supersolution to the parabolic equation on O_T is a function $\phi \in LSC(O_T)$ such that:

$a + F(t, x, \phi(t, x), p, X) \geq 0 \ \forall (t, x) \in O_T, \ (a, p, X) \in J_O^{2-}\phi(x)$.

More information can be found in the respective papers mentioned before.

Appendix 2: Matrix Notation for the Curvature

Carrying out the differentiations of the term $\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)$ we have

$$|\nabla\phi| \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = |\nabla\phi| \left[\left(\frac{\phi_x}{|\nabla \phi|} \right)'_x + \left(\frac{\phi_x}{|\nabla \phi|} \right)'_y \right] =$$

$$\frac{\phi_{xx}|\nabla\phi|^2 - (\phi_x^2\phi_{xx} + \phi_x\phi_y\phi_{xy})}{|\nabla\phi|^2} + \frac{\phi_{yy}|\nabla\phi|^2 - (\phi_y^2\phi_{yy} + \phi_y\phi_x\phi_{yx})}{|\nabla\phi|^2} =$$

$$\left(\phi_{xx} - \frac{(\phi_x^2\phi_{xx} + \phi_x\phi_y\phi_{xy})}{|\nabla\phi|^2} \right) + \left(\phi_{yy} - \frac{(\phi_y^2\phi_{yy} + \phi_y\phi_x\phi_{yx})}{|\nabla\phi|^2} \right) =$$

$$= trace \left(\begin{pmatrix} 1 - \frac{\phi_x^2}{|\nabla\phi|^2} & -\frac{\phi_x\phi_y}{|\nabla\phi|^2} \\ \frac{\phi_x\phi_y}{|\nabla\phi|^2} & 1 - \frac{\phi_y^2}{|\nabla\phi|^2} \end{pmatrix} \begin{pmatrix} \phi_{xx} & \phi_{xy} \\ \phi_{xy} & \phi_{yy} \end{pmatrix} \right). \tag{8.36}$$

In this way in the first matrix we deal with elements of the form

$$a_{ij}(|\nabla\phi|)\partial_{ij}\phi$$

[4] USC(O) is the set of upper semicontinuous functions on O

[5] LSC(O) is the set of lower semicontinuous functions on O

with $a_{ij} = \{a_{ij}(p)\} = \delta_{ij} - p_i p_j/|p|^2$ for $p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$. Using the notation $p \otimes p = \begin{pmatrix} p_1^2 & p_1 p_2 \\ p_1 p_2 & p_2^2 \end{pmatrix}$, for the classical Einstein sommation we can rewrite (8.36)

$$trace\bigg(\bigg(I - \frac{p \otimes p}{|p|^2}\bigg)X\bigg) = trace\bigg(A(p)X\bigg)$$

with $A(p) = I - \frac{p \otimes p}{|p|^2}$.

Appendix 3: Monotonically Inequality for Trace of Two Symmetric Matrices

Since X and Y are set of symmetric matrices we can write

$$Y \geq X \iff \forall\ \xi \in R^2 - \{0_{R^2}\},\ \xi^T Y \xi \geq \xi^T X \xi.$$

With the assumption that $p \neq 0$ to avoid the singularity, it can be easy proven that the matrix $A(p) = I - \frac{p \otimes p}{|p|^2}$ is positive with the following argument:

$$\forall\ \xi \in R^2 - \{0_{R^2}\}, \xi = \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} \quad \text{we have}$$

$$\xi^T A(p) \xi = \frac{1}{p^2}(\xi_1 p_2 - \xi_2 p_1)^2 \geq 0.$$

The matrix $A(p)$ is symmetric positive (for more details see [66]). Its eigenvalues are positive and there exists an orthonormal P an orthogonal matrix and D a diagonal matrix with positive values such that $D = P^T A P$. Thus one can write $A = GG^T$, with $G = PD^{1/2}$. With the following notations:

$$trace(AX) = trace(GG^T X) = trace(G^T X G)$$

$$trace(AX) = \sum_{i=1}^{2} G_i^T X G_i \text{ with } G_i \text{ the } i\text{th colomn of } G.$$

For $Y \geq X$ we can easily find

$$G_i^T X G_i \leq G_i^T Y G_i,\ \forall\ i\{1, 2\}.$$

The function $W(\mathbf{x})$ and the constant μ are positive brings to the following conclusion:

$$-\mu W(\mathbf{x}) trace(AY) \leq -\mu W(\mathbf{x}) trace(AX).$$

Appendix 4: Inequality Evaluation of $trace(A(p)X)+trace(A(p)Y)$

Before starting with the proof let's give some preliminary.

1. For $p, q \in R^n \backslash \{0\}$, it has been proven by [66] that

$$|\frac{p}{|p|} - \frac{q}{|q|}| \leq \frac{|p - q|}{\min(|p|, |q|)} = \rho(p, q)$$

2. The function $g : s \to \frac{1}{1+s^2}$ is bounded and $\sup |g'(s)|_{R^+} = \frac{3\sqrt{8}}{8}$ which proves that g is Lipschitz and one can write $\forall (s_1, s_2) \in (R^+)^2, |g(s_1) - g(s_2)| \leq \frac{3\sqrt{8}}{8}|s_1 - s_2|$. Considering $s = |\nabla u_0(x)|$ it can be shown [67] $|g(|\nabla u_0(x)|) - g(|\nabla u_0(y)|)| \leq k|x - y|$. The properties on $g\prime$ and $|\nabla u_0(y)|$ provides that $|\nabla g(|\nabla u_0(x)|)|$ is Lipschitz.

Let us first denote by $A(\mathbf{x}, p) = W(\mathbf{x})(I - \frac{p \otimes p}{|p|^2})$, a small modification of A. We have to remark is that $A(\mathbf{x}, p)$ has two eigenvalues 0 and and $W(\mathbf{x})$. The associated eigenvectors are respectively $\begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ and $\begin{pmatrix} -p_2 \\ p_1 \end{pmatrix}$. This means that it can be found an orthonormal matrix P such that:

$$A(p) = P(p) \begin{pmatrix} 0 & 0 \\ 0 & W(\mathbf{x}) \end{pmatrix} P(p)^T \text{ with } P(p) = \begin{pmatrix} \frac{p_1}{|p|} & -\frac{p_2}{|p|} \\ \frac{p_2}{|p|} & \frac{p1}{|p|} \end{pmatrix}$$

thus $trace(A(x,p)X) = trace(GG^TX) = trace(G^TXG)$ with $G = \begin{pmatrix} 0 & -\frac{p_2}{|p|}W(\mathbf{x}) \\ 0 & \frac{p1}{|p|}W(\mathbf{x}) \end{pmatrix}$. Moreover,

$$\begin{cases} trace(A(x,p)) = W(\mathbf{x}), \\ trace((G(p) - G(q))(G(p) - G(q))^T) = |\sqrt{W(\mathbf{x})}\frac{p}{|p|} - \sqrt{W(\mathbf{y})}\frac{q}{|q|}|^2 \end{cases} . \tag{8.37}$$

For all $r, s \in R^2$, the inequality (8.14) gives us:

$$(Xr, r) + (Ys, s) = r^TXr + s^TXs = (r^T, s^T) \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} (r, s) \leq$$

$$(r^T, s^T)\left[\mu_1 \begin{pmatrix} I & -I \\ -I & I \end{pmatrix} + \mu_2 \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}\right](r, s) = \mu_1\left[r^Tr - r^Ts - s^Tr + s^Ts\right] +$$

$$\mu_2\left[r^Tr + s^Ts\right] = \mu_1\left[(r-s)^T(r-s)\right] + \mu_2\left[|r|^2 + |s|^2\right] = \mu_1|r-s|^2 + \mu_2(|r|^2 + |s|^2) \tag{8.38}$$

Taking successively $r = G(x,p)e_i$ and $s = G(y,q)e_i$ with $(e_i)_i$ an orthonormal basis of R^2 (as done in [21], $A(x,p) = G(x,p)G^T(x,p)$), we get as well

$$trace(A(p)X) = \sum_{i=1}^{2}(XGe_i, Ge_i)$$

$$(Xr, r) = r^TXr = \sum (e_iG)^TXGe_i = trace(G^TXG) = trace(A(p)X).$$

Evaluating $trace(A(p)X) + trace(A(p)Y)$ and using the inequality (8.38) we have

$$trace(A(p)X) + trace(A(q)Y) \leq \mu_1 trace\left[(G(p)e_i - G(q)e_i)(e_i^TG^T(p) - e_i^TG^T(q))\right] +$$
$$\mu_2\left[G(p)e_i.(G(p)e_i)^T + G(q)e_i.(G(q)e_i)^T\right]$$

$$= \mu_1 trace\left[(G(p) - G(q))e_ie_i^T(G^T(p) - G^T(q))\right] + \mu_2\left[W(x) + W(y)\right]$$

$$= \mu_1 trace\Big[(G(p) - G(q))(G(p) - G(q))^T\Big] + \mu_2\Big[W(x) + W(y)\Big]$$

Hence

$$trace(A(p)X) + trace(A(q)Y) \leq \mu_1|\sqrt{W(x)}\frac{p}{|p|} - \sqrt{W(y)}\frac{q}{|q|}|^2 + 2\theta\mu_2 \tag{8.39}$$

since the function $W(x)$ is a bounded function by θ.

On the other hand, one has

$$|\sqrt{W(x)}\frac{p}{|p|} - \sqrt{W(y)}\frac{q}{|q|}|^2 = |\sqrt{W(x)}\frac{p}{|p|} - \sqrt{W(y)}\frac{p}{|p|} + \sqrt{W(y)}\frac{p}{|p|} - \sqrt{W(y)}\frac{q}{|q|}|^2 =$$

$$|(\sqrt{W(x)} - \sqrt{W(y)})\frac{p}{|p|} + \sqrt{W(y)}(\frac{p}{|p|} - \frac{q}{|q|})|^2 \leq 2(W(x) - W(y)^2 + 2W(y)|\frac{p}{|p|} - \frac{q}{|q|}|^2$$

Using the preliminary property we have

$$trace(A(p)X) + trace(A(q)Y) \leq \mu_1(2\xi|x - y|^2 + 8\theta\rho(p, q)^2) + 2\theta\mu_2$$

Chapter 9

Future Work

The ideas presented in this thesis can be expanded in different directions some of which are:

- Extensions of RHAM method to other image processing techniques such as debluring, inpaining and registration problems look promising. This will bring a comparison with high order models known with very interesting results but which suffers to have fast numerical methods for the solution.
- The improved selective segmentation model using one level-set and its viscosity solution introduced in Chapter 8 for 2-D images could be extended to 3-D with possible improvements to current 3-D result with dual level set technique. This shall be tested. Selective segmentation work is new to the literature. There are few published results and more work is needed to make it a very useful clinical tool.
- A hybrid model that combines registration and selective segmentation precesses could be a potential tool to improve segmentation results with a better and more accurate results.
- Using the graph cut theory to model the selective segmentation idea will be something totally new in literature. This work is currently ongoing.

Bibliography

[1] S. Abbasbandy. Solitary wave solutions to the kuramotosivashinsky equation by means of the homotopy analysis method. *Nonlinear Dynamics*, 52:35–40, 2008.

[2] S. Abbasbandy and F. Zakaria. Soliton solutions for the fifth-order kdv equation with the homotopy analysis method. *Nonlinear Dynamics*, 51(1):83 – 87, 2007.

[3] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[4] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6):641–647, 1994.

[5] G. Adomian. A review of the decomposition method and some recent results for nonlinear equations. *Comp and Math Appl*, 21:101 – 127, 1991.

[6] L. Alvarez, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29(3):845 866, 1992.

[7] I.V. Andrianov and V.V. Danishevky. Asymptotic approach for non-linear periodical vibrations of continous structures. *Journal of Sound and Vibration*, 249(3):465 – 481, 2002.

[8] Nguyen Thi Nhat Anh, Jianfei Cai, Juyong Zhang, and Jianmin Zheng. Robust interactive image segmentation using convex active contours. *IEEE Transactions on Image Processing*, 21(8):3734–3743, 2012.

[9] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama. Polynomial-time solutions to image segmentation. *Proc. of the 7th Ann. SIAM-ACM Conference on Discrete Algorithms*, pages 104–113, 1996.

[10] G. Aubert and P. Kornprobst. *Mathematical problems in image processing: Partial Differential Equations and the Calculus of Variations*. Springer, 2002.

[11] O. Axelsson and V.A. Barker. Finite element solution of boundary value problems. *Academic Press*, 1984.

[12] N. Badshah and K. Chen. Image selective segmentation under geometrical constraints using an active contour approach. *Commun. Comput. Phys.*, 7(4):759–778, 2009.

[13] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. *in IEEE ICCV*, pages 1–8, 2007.

[14] M. Barchiesi, S. H. Kang, T. M. Le, M. Morini, and M. Ponsiglione. A variational model for infinite perimeter segmentations based on lipschitz level set functions: Denoising while keeping finely oscillatory boundaries. *Multiscale Modeling & Simulation*, 8(5):1715–1741, 2010.

[15] G. Barles. Fully nonlinear neumann type boundary conditions for second-order elliptic and parabolic equations. *J. Differential Equations*, 106(1):90–106, 1993.

[16] G. Barles. Nonlinear neumann boundary conditions for quasilinear degenerate elliptic equations and applications. *J. Differential Eq.*, 34:191 224, 1999.

[17] G. Barles and P. L. Lions. Fully nonlinear neumann type boundary conditions for first-order hamilton-jacobi equations. *Nonlinear Anal.*, 16(2):143–153, 1993.

[18] I. Ben Ayed and A. Mitiche. A region merging prior for variational level set image segmentation. *Image Processing, IEEE Transactions on*, 17(12):2301 –2311, dec. 2008.

[19] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. *In Proc. International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation, Rennes*, 1997.

[20] R. P. Beyer and R. J. Leveque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal of Numerical Analysis*, 29(2):332–364, 1992.

[21] J. Biazar and B. Ghanbari. A new analytical approach for solving nonlinear boundary value problems in finite domains. *Applied Mathematics*, 2:987–992, 2011.

[22] X. Bresson, S. Esedoglu, P. Vandergheynst, J. P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.*, 28(2):151–167, 2007.

[23] X. Bresson, S. Esedoglu, P. Vandergheynst, J.P Thiran, and S. Osher. Global minimizers of the active contour/snake model. *CAM Report 05-04*, 2005.

[24] C. Brito-Loeza and K. Chen. Multigrid algorithm for high order denoising. *SIAM J.Imaging Sci.*, 3(3):363–389, 2010.

[25] R. L. Burden, , and J. D. Faires. *Numerical Analysis*. Brooks/Cole Pub Co; 6th edition, sixth edition edition, December 24, 1996.

[26] c. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*. Pitman Publishing, Marshfield, Massachusetts, 1984.

[27] V. Caselles, F. Catté, C. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer.Math.*, 66:1 31, 1993.

[28] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[29] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76:167–188, 1997.

[30] R. H. Chan and K. Chen. Multilevel algorithm for a poisson noise removal model with total-variation regularization. *International Journal of Computer Mathematics*, 84:1183–1198, 2007.

[31] T. F. Chan, S. Esedoglu, and M. Nikolov. Algorithm for finding global minimizers of image segmentation and denoising models. *UCLA CAM Report*, 04-54, 2004.

[32] T. F. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *Lecture Notes in Control and Information Sciences*, 219:241–252, 1996.

[33] T. F. Chan, B. Y. Sandberg, and L. A. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11:130–141, 2000.

[34] T. F. Chan, B.Y. Sandberg, and L. A. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11(2):130 – 141, 2000.

[35] T. F. Chan and J. Shen. *Image Processing and Analysis.* Society for Industrial and Applied Mathematics SIAM, 3600 University City Science Center, Philadelphia, PA 19104-2688, USA, 2005.

[36] T. F. Chan and J. H. Shen. *Image Processing and Analysis - Variational, PDE, Wavelet, and Stochastic Methods.* SIAM Publications, 2005.

[37] T. F. Chan and L. A. Vese. An active contour model without edges. In *Scale-Space*, pages 141–151, 1999.

[38] T. F. Chan and L. A. Vese. An efficient variational multiphase motion for the mumford-shah segmentation model. *Proc. Asilomar Conf. Signals, Systems, Computers*, 2000.

[39] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[40] T. F. Chan and L. A. Vese. Active contour and segmentation models using geometric pde's for medical imaging. In *Malladi, R. (Ed.), Geometric Methods in Bio-Medical Image Processing", Series: Mathematics and Visualization*, pages 63–75. Springer, 2002.

[41] H. Chang, Q. Yang, M. Auer, and B. Parvin. Modeling of front evolution with graph cut optimization. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 1, pages I –241 –I –244, 16 2007-oct. 19 2007.

[42] K. Chen. *Matrix Preconditioning Techniques and Applications.* Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU,UK, first edition, 2005.

[43] Y. G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *Journal of Differential Geometry*, 33(3):749 – 786, 1991.

[44] D. Chopp. Computing minimal surfaces via level set curvature flow. *J. Comput. Phys.*, 106:7791, 1993.

[45] N. Chumchob, K. Chen, and C. Brito-Loeza. Variational model for removal of combined additive and multiplicative noise. *International Journal of Computer Mathematics*, 2012.

[46] D. Cohen. On active contour models and ballons. *CVGIP: Image Understanding*, 53:211 218, 1991.

[47] S. D. Conte and C. W. De Boor. *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill Higher Education, 1980.

[48] M.G. Crandall, H. Ishii, and P.L. Lions. *User's Guide to Viscosity Solutions of Second Order Partial Differential Equations*, volume 27. American Mathematical Society, 1992.

[49] B. Dacorogna. Introduction to the calculus of varations. *Imperial College Press*, 2004.

[50] M. Dehghan, J. Manafian, and A. Saadatmandi. Solving nonlinear fractional partial differential equations using the homotopy analysis method. *Numerical Methods for Partial Differential Equations*, 26(2):448 479, 2010.

[51] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[52] H. Digabel and C. Lantuéjoul. Iterative algorithms. *In Actes du Second Symposium Européen d'Analyse Quantitative des Microstructures en Sciences des Matériaux, Biologie et Médecine, Caen, 4-7 October 1977 (1978), J.-L. Chermant, Ed., Riederer Verlag, Stuttgart, pp. 85 - 99.*

[53] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, 1996.

[54] L. C. Evan and J. Spruck. Motion of level sets by mean curvature i. *Journal of Differential Geometry*, 33(3):635–681, 1991.

[55] L. C. Evan and J. Spruck. Motion of level sets by mean curvature ii. *Transaction of the American Mathematical Society*, 330:321–332, 1992.

[56] L. C. Evan and J. Spruck. Motion of level sets by mean curvature iii. *Journal of Geometric Analysis*, pages 121–150, 1992.

[57] N. Forcadel, C. Le Guyader, and C. Gout. Image segmentation using a generalized fast marching method. *Numerical Algorithms*, 48(2):189–212, 2008.

[58] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, 1963.

[59] B. Ghanbari, L. Rada, and K. Chen. A restarted iterative homotopy analysis method for two nonlinear models from image processing. *International Journal of Computer Mathematics*, 2012, submitted.

[60] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I, The Lagrangian Formalism*. Springer-Verlag, 1996.

[61] M. Giaquinta and S. Hildebrandt. *Calculus of Variations II, The Hamiltonian Formalism*. Springer-Verlag, 1996.

[62] Y. Giga and M.-H. Sato. Generalized interface evolution with neumann boundary condition. *Proc. Japan Acad. Ser. A*, 67:263–266, 1991.

[63] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation. Monographs in Mathematics.* Birkhuser Boston, 1984.

[64] G. Golub and C. Van Loan. *Matrix Computations.* The Johns Hopkins University Press; 3rd edition, Baltimore, USA, 1996.

[65] C. Gout. Viscosity solutions for geodesic active contour under geometrical conditions. *International Journal of Computer Mathematics*, 85(9):1375 1395, 2008.

[66] C. Gout and C. Le Guyader. Segmentation of complex geophysical structures with well data. *Computational Geosciences*, 10(4):361–372, 2006.

[67] C. Gout, C. Le Guyader, and L. A. Vese. Segmentation under geometrical consitions with geodesic active contour and interpolation using level set methods. *Numerical Algorithms*, 39:155–173, 2005.

[68] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768 1783, 2006.

[69] C. Le Guyader and C. Gout. Geodesic active contour under geometrical conditions theory and 3d applications. *Numerical Algorithms*, 48:105–133, 2008.

[70] J. Hadamard. Sur les problmes aux drives partielles et leur signification physique. page 49 52, 1902.

[71] Steve Hanov. Wavelets and edge detection. 2006.

[72] T. Hayat, N. Ahmed, M. Sajid, and S. Asghar. On the mhd flow of a second grade fluid in a porous channel. *Computers and Mathematics with Applications*, pages 407–414, 2007.

[73] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.

[74] S. Hildebrandt and A. Tromba. *The Parsimonious Universe: Shape and Form in the Natural World.* Springer, 1 edition, July 18, 1996.

[75] W. Hinterberger, Hintermüller M., Kunisch K., M. Von Oehsen, and O. Scherzer. Tube methods for bv regularization. *J. Math. Imaging Vis.*, 19(3):219–235, 2003.

[76] M. Hintermuller and W. Ring. An inexact newton-cg-type active contour approach for the minimization of the mumford-shah functional. *J. Math. Imaging Vis.*, 20:19–42, 2004.

[77] M. Hintermüller and G. Stadler. An infeasible primal-dual algorithm for total bounded variation–based inf-convolution-type image restoration. *SIAM J. Sci. Comput.*, 28(1):1–23, 2006.

[78] H. Ishii and M.-H. Sato. Nonlinear oblique derivative problems for singular degenerate parabolic equations on a general domain. *Nonlinear Anal.*, 57:10771098, 2004.

[79] H. Jafari and S. Seifi. Solving a system of nonlinear fractional partial differential equations using homotopy analysis method. *Communications in Nonlinear Science and Numerical Simulation*, 14(5):1962 – 1969, 2009.

[80] M. Jeon, M. Alexander, W. Pedrycz, and N. Pizzi. Unsupervised hierarchical image segmentation with level set and additive operator splitting. *PRL*, 26(10):1461–1469, July 2005.

[81] G. S. Jiang and D. Peng. Weighted eno schemes for hamilton-jacobi equations. *Journal of Scientific Computing*, 21(6):2126–2143, 1999.

[82] W. Kahan. *Gauss-Seidel Methods of Solving Large Systems of Linear Equations.* PhD thesis, University of Toronto, Canada, 1958.

[83] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Jurnal of Computer Vision*, 1(4):321–331, 1988.

[84] S. Kichenassamy, A. Kumar, P. Olver, and A. Yezzi. Snake: Conformal curvature flows: From phase transitions to active vision. *Archive Rational Mech. Anal.*, 134(3):275–301, 1996.

[85] J. Kim, III Fisher, J.W., A. Yezzi, M. Cetin, and A.S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *Image Processing, IEEE Transactions on*, 14(10):1486 –1502, 2005.

[86] K. Königsberger. Analysis 2. *Springer*, 2004.

[87] G. Landl and R. S. Anderssen. Non-negative differentially constrained entropy-like regularization. *Inverse Problems*, 12:35–53, 1996.

[88] L. P. Lebedev and I. I. Vorovich. *Functional Analysis in Mechanics.* Springer-Verlag, 2002.

[89] D. Lesnic. Characterizations of the functions with bounded variations. *Acta Universitatis Apulensis*, 6:47–54, 2003.

[90] C. Li, C. Kao, J. Gore, and Z. Ding. Implicit active contours driven by local binary fitting energy. pages 1–7, 2007.

[91] C. Li, C. Kao, J. C. Gore, and Z. Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.*, 17(10):1940–1949, October 2008.

[92] C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: A new variational formulation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:430–436, 2005.

[93] S. Liang and D. J. Jeffrey. An analytical approach for solving nonlinear boundary value problems in finite domains. *Numerical Algorthms*, 56:987–992, 2010.

[94] S. J. Liao. *The proposed homotopy analysis technique for the solution of nonlinear problems.* Ph.D. Thesis, Shanghai Jiao Tong University, 1992.

[95] S. J. Liao. A challenging nonlinear problem for numerical techniques. *J. Comput. Appl. Math.*, 181:467–472, 2005.

[96] S. J. Liao. A new branch of solutions of boundary-layer flows over a permeable stretching plate. *International Journal of Non-Linear Mechanics*, 42(6):819 – 830, 2007.

[97] S. J. Liao. Notes on the homotopy analysis method: Some definitions and theorems. *Communications in Nonlinear Science and Numerical Simulation*, 14(4):983 – 997, 2009.

[98] S. J. Liao. On the relationship between the homotopy analysis method and euler transform. *Communications in Nonlinear Science and Numerical Simulation*, 15(6):1421 – 1431, 2010.

[99] S.J. Liao. Beyond perturbation: Introduction to the homotopy analysis method. 2003.

[100] S.J Liao. On the homotopy analysis method for nonlinear problems. *Applied Mathematics and Computation*, 147(2):499 – 513, 2004.

[101] S.J Liao and Y. Tan. A general approach to obtain series solutions of nonlinear differential equations. *Stud. Appl. Math.*, 119:297 – 355, 2007.

[102] J. Lie, M. Lysaker, and X. C. Tai. A variant of the level set method and applications to image segmentation. *Mathematics of Computation*, 75(255):1155–1174, 2006.

[103] P. L. Lions. Neumann type boundary conditions for hamilton-jacobi equations. *Duke Math. J.*, 52:793–820, 1985.

[104] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equationss. *Appl. Math. Lett.*, 4(2):25–29, 1991.

[105] T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method for partial differential equations and its application to navier-stokes equations. *RAIRO Math. Model. and Numerical Analysis*, 26(6):673–708, 1992.

[106] T. Lu, P. Neittaanmaki, and X. C. Tai. A parallel splitting-up method for partial differential equations and its application to navier-stokes equations. *RAIRO Mathematical Modelling and Numerical Analysis*, 26(6):673–708, 1992.

[107] J. Malik, Th. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43:7–27, June 2001.

[108] R. Malladi, Sethian, J., and B. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:158–175, 1995.

[109] S. Mallat. *A wavelet tour of signal processing*. Academic Press, USA, 1998.

[110] O. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the bhattacharyya gradient flow. *Image Processing, IEEE Transactions on*, 16(11):2787–2801, 2007.

[111] A. Mitiche and I. Ben-Ayed. *Variational and Level Set Methods in Image Segmentation*. Springer Topics in Signal Processing. Springer, 2010.

[112] J. M. Morel and S. Solimini. Variational methods in image segmentation: A constructive approach. *Revista Matematica Universidad Complutense de Madrid*, 1:169–182, 1988.

[113] J. M. Morel and S. Solimini. *Variational methods in image segmentation.* Birkhauser Boston Inc., Cambridge, MA, USA, 1995.

[114] W. Mulder, S. Osher, and J. Sethian. Computing interface motion in compressible gas dynamics. *J. Comput. Phys.*, 100:209228, 1992.

[115] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure Applied Mathematics*, 42:577–685, 1989.

[116] J. A. Murdock. *Perturbations: theory and methods.* New York: John Wiley and Sons, 1991.

[117] A. H. Nayfeh. *Perturbations methods.* New York: John Wiley and Sons, 2000.

[118] A. M. Oberman. A convergent difference scheme for the infinity laplacian: construction of absolutely minimizing lipschitz extensions. *Math. Comput.*, 74(251):1217–1230, 2005.

[119] M. Von Oehsen. *Multiscale Methods for Variational Image Processing.* Logos Verlag Berlin, Comeniushof, Gubener str. 47, 10243 Berlin, first edition, 2002.

[120] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, USA, 1970.

[121] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, New York, 2003.

[122] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[123] S. Osher and C. W Shu. High order essentially non-oscillatory schemes for hamilton-jacobi equaitons. *Journal of Computational Physics*, 28(4):907–922, 1991.

[124] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Comput. Vision*, 46(3):223–247, 2002.

[125] D. Peng, B. Merriman, S. Osher, H. K. Zhao, and M. Kang. A pde based fast local level set method. *J. Comput. Phys.*, 155:410438, 1999.

[126] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[127] P. Perona, T. Shiota, and J. Malik. Anisotropic diffusion. *B.M. ter Haar Romeny (Ed.), Geometry Driven Diffusion in Computer Vision*, Chapter 2:72–92, 1994.

[128] H. A. Priestley. *Introduction to integration.* Oxford University Press, 1997.

[129] L. Rada and K. Chen. A new variational model with dual level set functions for selective segmentation. *Commun. Comput. Phys.*, 12(1):261–283, 2012.

[130] L. Rada, B. Ghanbari, and K. Chen. A restarted iterative homotopy analysis method for three-dimensional image segmentation. *International Journal of Signal and Imaging Systems Engineering*, 2012, to appear.

[131] R. D. Richtmyer and K. W. Morton. *Difference methods for initial-value problems.* Second edition. Interscience Tracts in Pure and Applied Mathematics, No. 4. Interscience Publishers John Wiley & Sons, Inc., New York-London-Sydney, 1967.

[132] R. Ronfard. Region-based strategies for active contour models. *Int. J. of Comput. Vis.*, 13:229–251, 1994.

[133] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *in ACM SIGGRAPH*, 2004.

[134] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal*, 29:867 884, 1992.

[135] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[136] L.I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 31 –35 vol.1, nov 1994.

[137] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics; 2nd edition, PA, USA, 2003.

[138] H. Sagan. *Introduction to the Calculus of Variations.* 1992.

[139] M. Sajid, T. Hayat, and S. Asghar. Non-similar solution for the axisymmetric flow of a third-grade fluid over a radially stretching sheet. *Acta Mechanica*, 189:193–205, 2007.

[140] Ch. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A level set model for image classification. *International Journal of Computer Vision*, 40:187–197, 2000. 10.1023/A:1008183109594.

[141] G. Sapiro. *Geometric partial differential equations and image analysis.* Cambridge University Press, New York, NY, USA, 2001.

[142] G. Sapiro, R. Kimmel, and V. Casells. Object detection and measurements in medical images via geodesic deformable contours. In R. A. Melter, A. Y. Wu, F. L. Bookstein, & W. D. Green, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2573 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 366–378, 1995.

[143] J. Savage and K. Chen. An improved and accelerated non-linear multigrid method for total-variation denoising. *International Journal of Computer Mathematics*, 82(8):1001–1015(15), 2005.

[144] J. Savage and K. Chen. On multigrids for solving a class of improved total variation based staircasing reduction models. *Image Processing Based On Partial Differential Equations, eds. X.-C. Tai, K.-A. Lie, T. F. Chan and S. Osher*, 82:69–94, 2006.

[145] M. Schechter. *Principles of Functional Analysis*. AMS, 2nd edition, 2001.

[146] D. Sen and S. K. Pal. Histogram thresholding using fuzzy and rough measures of association error. *Image Processing, IEEE Transactions on*, 18(4):879–888, 2009.

[147] J. A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101:487–499, 1985.

[148] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999.

[149] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[150] J. A. Sethian. *Level Set Methods and Fast Marching Methods:* Evolving interfaces in computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences. Cambridge University Press, The Edinburgh Building,Cambridge, UK, 2002.

[151] J. A. Sethian and R. Fedkiw. Level set methods: *An Overview and Some Recent Results. J. Comp. Phys.*, 169:463502, 2001.

[152] L. G. Shapiro and G. C. Stockman. Computer vision. *New Jersey, Prentice-Hall, ISBN 0-13-030796-3*, pages 279–325, 2001.

[153] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Chapman & Hall, 2 edition, 1998.

[154] D. M. Strong and Tony F. Chan. Edge preserving and scale dependant properties of total variation regularization. *Inverse Problems*, 19:165–187, 2003.

[155] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Phys.*, 114:146–159, 1994.

[156] X. Ch. Tai, O. Christiansen, P. Lin, and I. Skjaelaaen. Image segmentation using some piecewise constant level set methods with mbo type of projection. *Int. J. Comput. Vision*, 73(1):61–76, 2007.

[157] A. N. Tikhonov. On the solution of ill-posed problems and the method of regularization. *Dokl. Akad. Nauk SSSR*, 151:501504, 1963 (Russian).

[158] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963 (Russian).

[159] A. N. Tikhonov and V.Y. Arsenin. Solutions of ill-posed problems. *New York: Winston*, 1977.

[160] Ulrich Trottenberg and Anton Schuller. *Multigrid*. Academic Press, Inc., Orlando, FL, USA, 2001.

[161] A. Tsai, A. J. Yezzi Jr., and A. S. Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*.

[162] N. Valliammal and S. N. Geethalakshmi. Performance analysis of various leaf boundary edge detection algorithms. In *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India*, A2CWiC '10, pages 34:1–34:6, New York, NY, USA, 2010. ACM.

[163] R. A. Varga. Matrix iterative analysis. 1992.

[164] L. A. Vese. Multiphase object detection and image segmentation. *UCLA CAM Report*, 02-36, 2002.

[165] L. A. Vese. *Variational Methods in Image Processing*. Chapman & Hall, 2012.

[166] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50:271–293, 2002.

[167] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:583–598, 1991.

[168] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[169] J. Weickert. A review of nonlinear diffusion filtering. In Bart ter Haar Romeny, Luc Florack, Jan Koenderink, and Max Viergever, editors, *Scale-Space Theory in Computer Vision*, volume 1252 of *Lecture Notes in Computer Science*, pages 1–28. Springer Berlin / Heidelberg, 1997.

[170] J. Weickert. Anisotropic diffusion in image processing, 1998.

[171] J. Weickert and G. Kühne. Fast methods for implicit active contour. In N. Paragios S. Osher, editor, *Geometric Level Set Methods in Imaging, Vision, and Graphics*, pages 43–57. Springer New York, New York, 1995.

[172] J. Weickert and G. Kühne. Fast methods for implicit active contours models, preprint 61. *Universität des Saarlandes, Saarbrücken*, 2002.

[173] J. Weickert, B. M. ter Haar Romeny, and Max A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *Scale-Space theory in computer vision, Lecture Notes in Computer Science*, 1252:260–271, 1997.

[174] J. Weickert, Bart M. Ter Haar Romeny, and Max A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.

[175] J. Weickert, B.M. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *Image Processing, IEEE Transactions on*, 7(3):398–410, 1998.

[176] X. Xie and M. Mirmehdi. Radial basis function based level set interpolation and evolution for deformable modelling. *Image and Vision Computing*, 29(3):167–177, February 2011.

[177] A. Histace Fr*acuteee*édéric. Precioso J. Kilgallon Y. Zhang, B. J. Matuszewski and C. Moore. Boundary delineation in prostate imaging using active contour segmentation method with interactively defined object regions. *Int J. Comput. Math.*, 6367(3), 2010.

[178] F. Yang, K. Chen, and B. Yu. Homotopy method for a mean curvature-based denoising model. *Applied Numerical Mathematics*, 62(3):185 – 200, 2012.

[179] J.P. Zhang, K. Chen, and B. Yu. Local information based image selection segmentation model and fast robust algorithm. to appear.

[180] J.P. Zhang, K. Chen, and B. Yu. 3d mathematical models of image segmentation and fast robust algorithms. *Int J. Comput. Math.*, 29(3), 2011. to appear.

[181] K. Zhang, H. Song, and L. Zhang. Active contours driven by local image fitting energy. *Pattern Recognition*, 43(4):1199 – 1206, 2010.

[182] H. K. Zhao, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:79–195, 1996.

[183] H. Zhu, H. Shu, and M. Ding. Numerical solutions of partial differential equations by discrete homotopy analysis method. *Applied Mathematics and Computation*, 216(12):3592 – 3605, 2010.

[184] S.-P. Zhu. An exact and explicit solution for the valuation of american put options. *Quantitative Finance*, 6(3):229–242, 2006.

[185] W. Zhu and T. F. Chan. Image denoising using mean curvature. *Unpublished article (http://www.math.nyu.edu/∼wzhu/)*, 2008.

[186] W. Zhu and T. F. Chan. Image denoising using mean curvature of image surface. *SIAM J. Imaging Sci.*, 5:1–32, 2012.

[187] W. P. Ziemer. *Weakly Differentiable Functions*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[188] S. W. Zucker. Region growing: childhood and adolescence. *Computer Graphics and Image processing*, 5:382–399, 1976.